Color	Red	Green	Blue
Black	0	0	0
White	2	2	2
Red	2	0	0
Green	0	2	0
Blue	0	0	2
Yellow	2	2	0
Magenta	2	0	2
Cyan	0	2	2

Table 21 Minimum Color List Table

When a decoder supporting this Minimum Color List receives an RGB value not in the list, it will map the received value to one of the values in the list via the following algorithm:

- All one (1) values are to be changed to 0
- All two (2) values are to remain unchanged
- All three (3) values are to be changed to 2

For example, the RGB value (1,2,3) will be mapped to (0,2,2), (3,3,3) will be mapped to (2,2,2) and (1,1,1) will be mapped to (0,0,0).

Table 22 is an alternative minimum color list table supporting 22 colors.

Color	Red	Green	Blue
Black	0	0	0
Gray	1	1	1
White	2	2	2
Bright White	3	3	3
Dark Red	1	0	0
Red	2	0	0
Bright Red	3	0	0
Dark Green	0	1	0
Green	0	2	0
Bright Green	0	3	0
Dark Blue	0	0	1
Blue	0	0	2
Bright Blue	0	0	3
Dark Yellow	1	1	0
Yellow	2	2	0
Bright Yellow	3	3	0
Dark Magenta	1	0	1
Magenta	2	0	2
Bright Magenta	3	0	3
Dark Cyan	0	1	1
Cyan	0	2	2
Bright Cyan	0	3	3

Table 22 Alternative Minimum Color List Table

When a decoder supporting the Alternative Minimum Color List in Table 22 receives an RGB value not in the list (i.e., an RGB value whose non-zero elements are not the same value), it will map the received value to one of the values in the list via the following algorithm:

- For RGB values with all elements non-zero and different - e.g., (1,2,3), (3,2,1), and (2,1,3), the 1 value will be changed to 0, the 2 value will remain unchanged, and the 3 value will be changed to 2.
- For RGB values with all elements non-zero and with two common elements - e.g. (3,1,3), (2,1,2), and (2,2,3), if the common elements are 3 and the uncommon one is 1, then the 1 elements is changed to 0; e.g. (3,1,3) -> (3,0,3). If the common elements are 1 and the uncommon element is 3, then the 1 elements are changed to 0, and the 3 element is changed to 2; e.g. (1,3,1) -> (0,2,0). In all other cases, the uncommon element is changed to the common value; e.g., (2,2,3) -> (2,2,2), (1,2,1) -> (1,1,1), and (3,2,3) -> (3,3,3).

All decoders not supporting either one of the two color lists described above, must support the full 64 possible RGB color value combinations.

9.21 Character Rendition Considerations

In NTSC Closed Captioning, decoders were required to insert leading and trailing spaces on each caption row. There were two reasons for this requirement:

1. to provide a buffer so that the first and last characters of a caption row do not fall outside the safe title area, and

2. to provide a black border on each side of a character so that the "white" leading pixels of the first character on a row and the trailing "white" pixels of the last character on a row do not bleed into the underlying video.

Since caption windows are required to reside in the safe title area of the DTV screen, reason number 1 (above) is not applicable to DTVCC captions.

The attributes available in the **SetPenAttributes** command for character rendition (e.g.,character background and edge attributes) provide unlimited flexibilty to the caption provider when describing caption text in an ideal decoder implementation. However, manufacturers need only implement a minimum of pen attributes and font styles. Thus it is recommended that no matter what the level of implementation, decoder manufacturers should take into account the readablility of all caption text against a variety of all video backgrounds, and should implement some automatic character delineation when the individual control of character foreground, background and edge is not supported; and when only a minimum number of font styles are implemented.

9.22 DTVCC Section 8.9 - Service Synchronization

Service Input Buffers must be at least 128 bytes in size. Caption providers must keep this lower limit in mind when following Delay commands with other commands and window text. In other words, no more than 128 bytes of DTVCC commands and text should be transmitted (encoded) before a pending Delay command's delay interval expires.

9.23 DTV to NTSC Transcoders

It is anticipated that receiver (decoder) manufacturers will develop devices (e.g., settop boxes) which process an DTV stream and transcode it for display on NTSC monitors. The DTVCC command set is not necessarily transcodable to NTSC captions; i.e., there are DTVCC captions which have no NTSC equivalent.

Although receiver manufacturers are free to attempt an automatic transcode of the captions, there is no guarantee that the captions will appear as the caption provider intended. Caption providers apply many techniques to make the captions easy to read and as unobtrusive as possible over the underlying video. To maintain caption quality during an automated transcode process, a set of conversion rules would have to be defined which cover all possible window, pen and text attribute combinations.

Therefore, a separate NTSC caption channel was added to the Picture User Data (see Section 4.3). This channel allows caption providers to encode dual caption streams within the same programming. NTSC captions are under the complete control of the caption provider; and thus, no automated transcoding of captions is necessary.

10 DTVCC Authoring and Encoding for Transmission

This section describes a DTV captioning "food chain". This is the path the captioning information takes from initial authoring intentions to being multiplexed into the ATSC emission bit stream. The content of this section is an example for information.

10.1 Caption Authoring and Encoding

High quality captioning starts with the creation of the captioning intentions. This is a high level, generally editable, representation of how and when the captions should appear when rendered on the consumer receiver. SMPTE 12M time code is generally used for synchronization with picture. The output of the initial authoring process is generally a computer file that contains a list of time codes and the intention as to what the receiver should render when the picture, with the corresponding time code, appears on the display device. This computer file is typically editable. The file may be stored on a hard disc or floppy disc, and distributed by either computer networking techniques, or via floppy net. This process is illustrated in Figure 19.

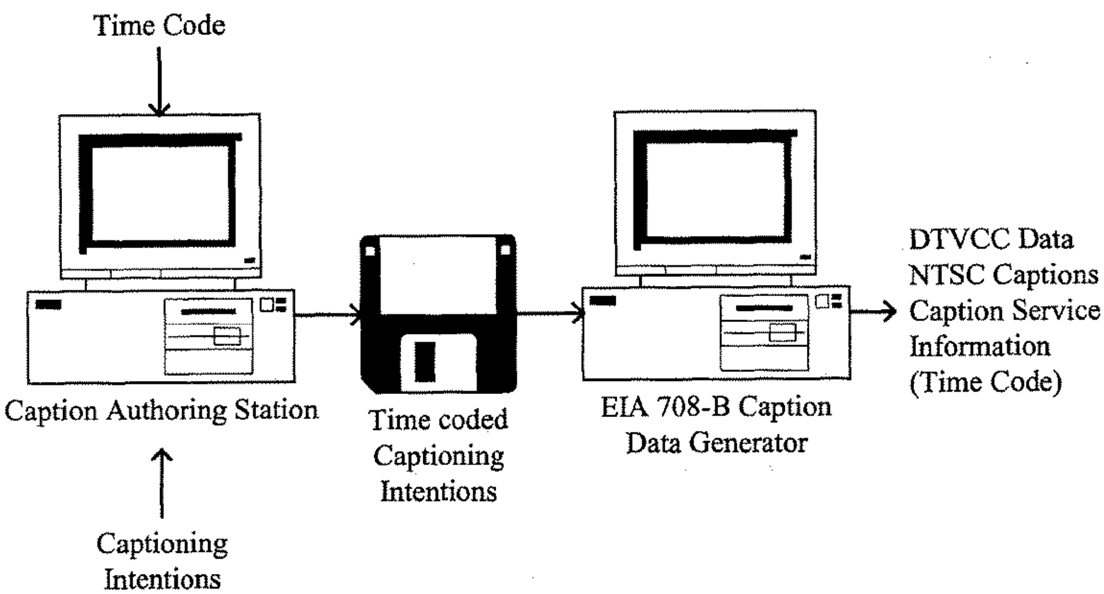

Figure 19 Caption Authoring and Encoding into Caption Channel Packets

The caption intentions are rendered into variable length *DTV Caption Channel Packets* (see Section 5). This rendering process is performed by an EIA-708-B caption data generator. This generator may also create a EIA-608-A (NTSC) captioning stream. This rendering process should consider the latency of the caption decoding process in the receiver, and thus should generate the packets pre-timed for transmission slightly early. Besides rendering the intentions into DTV Caption Channel Packets, the generator should create the caption service information that will be used to create the caption service descriptors that are carried in both the MPEG-2 PMT and in the ATSC PSIP EIT.

When delivered to the consumer receiver by the ATSC DTV system, the NTSC and DTV caption data is carried in the MPEG-2 picture user data, where a fixed number of bytes is allocated for each frame. The allocation provides 9,600 bits per second of data capacity. The number of bytes carried with each picture frame is dependent on the picture frame rate (see table 3 in section 4.4.2). In the case of 29.97 or 30 fps pictures, there are 4 bytes of NTSC

caption data and 36 bytes of DTV caption data carried per frame. The data rate of DTV Caption Channel Packets is equal to or lower than 9600 bps and so some zero padding is generally required. After the variable length DTV Caption Channel Packets are formed, it is necessary to parse these packets into the fixed length blocks that will be included with individual picture frames in the final emission multiplex. Many of these blocks will include zero padding so that the occupied bit rate is padded out to the full 9600 bps rate. This process is illustrated in figure 10-2.

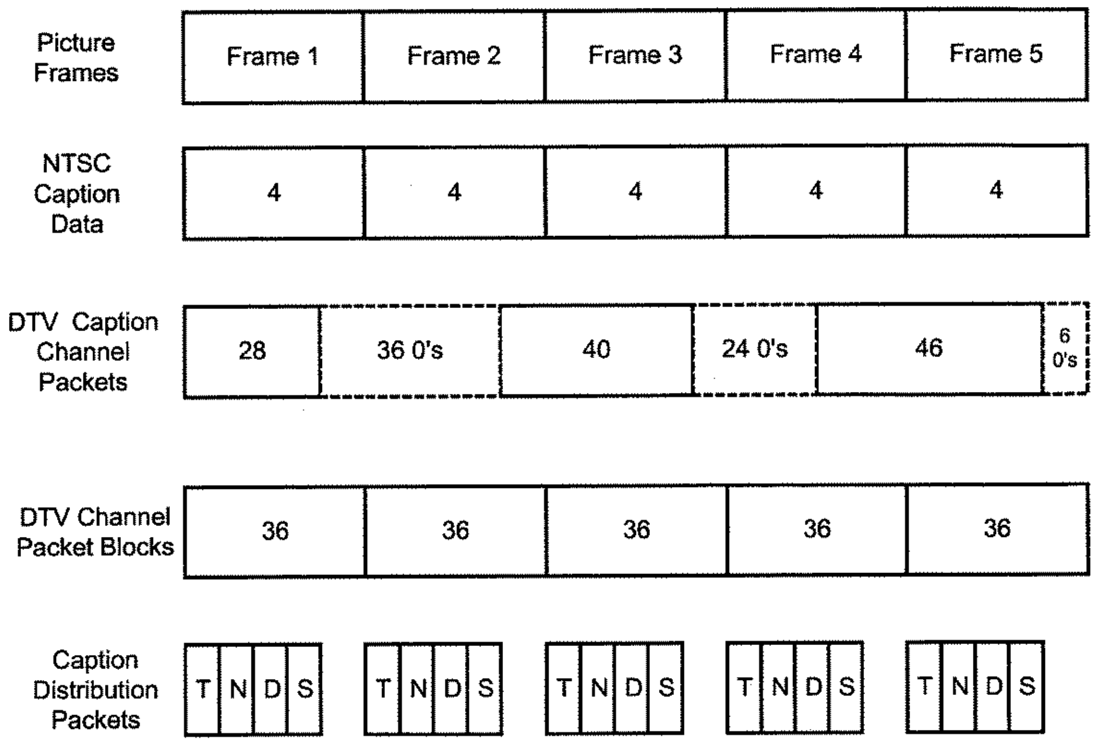

Figure 20 Relationship Between Caption Data and Picture Frames

Exhibit 15 Page 283

SONY0005800

Figure 20 illustrates five picture frame periods, at a 29.97/30 Hz frame rate. At this frame rate there are 4 bytes of NTSC caption data per frame. Three DTV Caption Channel Packets have been generated during the time of these 5 frames. These variable length packets have lengths of 28, 40, and 46 bytes. The DTV caption data is then formed into fixed length DTV Channel Packet Blocks that will be included in the corresponding MPEG-2 picture user data (see section 4) area. In this 29.97/30 Hz frame rate example, these blocks have a uniform length of 36 bytes. The first of these blocks contain the 28 bytes of the first DTV Caption Channel Packet plus 8 bytes of zero padding. The second block contains 28 bytes of zero padding followed by the first 8 bytes of the second DTV Caption Channel Packet, and so on. The result of this process is 36 bytes of DTV captioning data per picture frame. The final line in this figure shows the formation of all of the captioning data into *Caption Distribution Packets* that will be described in Section 11. These packets have a 1:1 relationship to video frames, and can include time code (T), the NTSC caption data (N), the blocked and zero padded DTV caption data (D), and the caption service information (S).

10.2 Monitoring Captions

Caption data may be monitored at various points during the distribution chain. To monitor the captions as they will be displayed on a consumer receiver, and to evaluate video/caption synchronization, it is necessary to decode the caption data after it has been rendered into a form (such as the Caption Distribution Packet) where there is an association of sets of caption data bytes with particular video frames.

10.3 Encoder Interfacing

DTVCC and NTSC captions are multiplexed into the Picture User Data in the MPEG-2 video elementary stream. This multiplexing is generally done within the MPEG-2 video encoder and so the caption data must be delivered to this encoder. A Caption Service Descriptor is included in the PMT and in the EIT and so caption service information must be delivered to the PSIP generator, and to the MPEG-2 PMT generator. (It is possible that both PMT and EIT would be created by a common functional block.) It is likely that all of the caption information will arrive at the encoding system on a single interface connection, and be distributed to the various destinations by looping the interface connection to multiple functional blocks. See Figure 21.

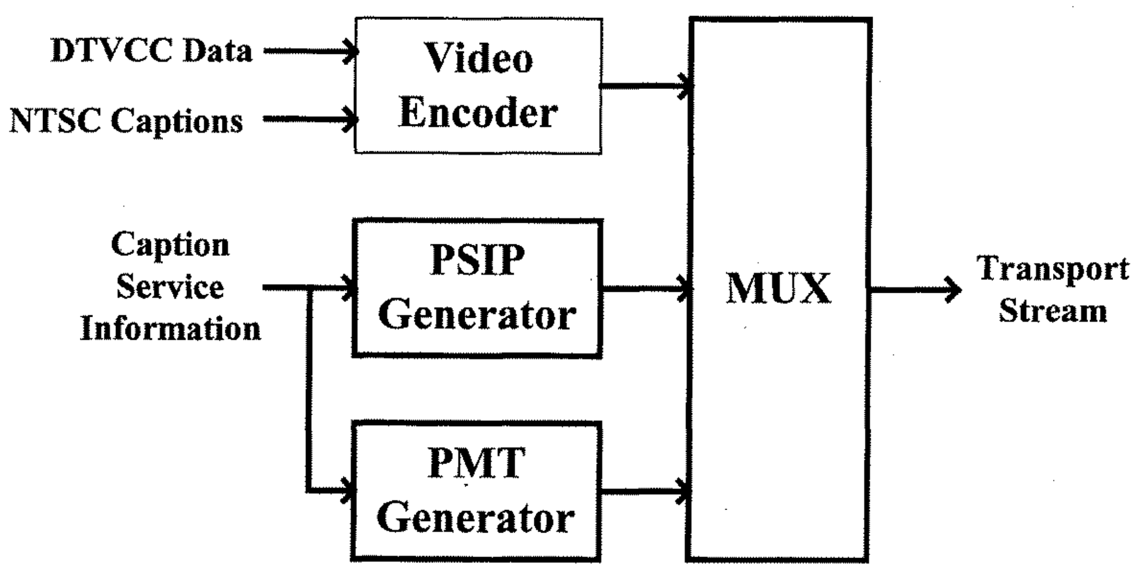

Figure 21 Interface of Caption Data to ATSC Emission Encoding Equipment

Several types of interfaces for the caption data may exist. These include a serial data format carried on an asynchronous RS232 like interface, embedding into an AES3 (digital audio interface) data stream, or embedding into SMPTE 259M or 292M serial digital video streams. For purposes of interoperability it is useful if there is as much commonality as possible to the data format on these different types of interfaces. The Caption Distribution Packet (CDP) described in Section 11 can carry DTVCC data, NTSC captions, caption service information, and time code.

Section 11 defines an asynchronous serial interface to carry the CDP. SMPTE is standardizing carriage of the CDP over AES3, 259M, and 292M interfaces.

11 DTV Closed Captioning Content Package

This section describes how to create a closed captioning content package, designated the Caption Distribution Packet (CDP), which may contain time code, EIA-708-B closed captioning data, and ATSC closed caption descriptor information. The CDP is suitable for transport over a variety of streaming interfaces. The details of one specific serial interface are defined.

11.1 Introduction

The process of creating and delivering closed captions for ATSC DTV involves authoring captions into a representation which represents the frame accurate captioning intentions (e.g. SAMI high level representation), rendering the intentions into the EIA-708-B caption syntax, transport of this EIA-708-B data via storage and/or streaming media to the point of emission, and then packaging the EIA-708-B data into the MPEG-2 picture user data as specified in ATSC A/53. During this process the caption data must be kept properly synchronized to the picture and sound. It is also necessary to create, transport, and include the caption service information in order to create the Caption Service Descriptor that is carried in the PMT and EIT tables in the MPEG-2 emission transport stream. During the distribution chain, the EIA 708-B captioning data may be rendered and displayed for quality control purposes.

This section describes how to create a captioning content package, consisting of a defined sequence of bytes, which may carry the following:

- time code;
- EIA-708-B caption;
- EIA-608-A NTSC caption data;
- caption service information to form the caption service descriptor; and/or
- sequence counts to detect discontinuities in the stream of caption data packets.

The CDP may be employed in various types of transport of captioning data, including file formats, RS232, TIA/EIA-574, AES3, SMPTE 259M, SMPTE 292M, etc. The CDP may be further encapsulated into a SMPTE defined Key-Length_Value (KLV) construct. Details on the KLV construct may be obtained from SMPTE.

11.1.1 Frame Rates

The transport of closed captioning information over the DTV emission system involves packaging the CC data into the MPEG-2 picture user data area. In order to do this the captioning data must be packaged at the same frame rate as that used by the video encoder. Rendering the captioning data into a particular frame rate is done prior to or during creation of the CDP. If the video encoder encodes at a frame rate that differs from the frame rate of the CDP, the captioning data must be re-framed. This can be done by the video encoder, by a caption data server, or by other equipment upstream of the video encoder. However, if the video encoder is responsible for the determination of the encoded picture frame rate, the re-framing should be done in the video encoder or in a captioning server tightly coupled (with 2-way communication) to the video encoder.

From the point of view of captioning, frame rates which differ by 0.1% may be considered identical. For example, if picture and captions are rendered at a 30 Hz frame rate, they may both be played at 29.97 Hz without any reframing of the caption data. As long as captions are delivered at the same rate as the picture, and this rate does not change by more than 0.1 %, no reframing is needed. Also, captions may be generated at a 25 or nominal 30 Hz rate, and used with 50 or nominal 60 Hz video. In this case, a video encoder should place the first half of the caption data in the first picture user data area, and place the second half of the caption data in the second picture user data area.

11.1.2 Time Code

The CDP may carry a time code which may be derived from SMPTE 12M VITC or LTC. Carriage of a time code provides an important tool to allow captions to be kept properly synchronized with pictures. The picture, sound, and caption data elements may flow through differing paths to the emission encoding and multiplexing equipment. The

inclusion of time code within each type of element makes it possible for the final multiplexer to provide buffering in order to deliver a properly synchronized multiplex of elements to the final receiver.

11.1.3 Caption Data

The caption data contained in the CDP is fully framed and formatted for direct inclusion within the ATSC video elementary stream picture user data, as defined in ATSC A/53, sections 5.2.2 and 5.2.3.

11.1.4 Caption Service Information

The caption service information in the CDP is fully framed and formatted for direct inclusion within the caption service descriptor as defined in ATSC A/65, section 6.7.3, table 6.17.

To reduce the data rate to carry the CDP stream (see section 11.1.5 below), the caption service information may be spread over a sequence of CDP packets. The receiver of the CDP stream must be able to collect service information from a sequence of CDP packets, and must be able to detect when the service information has changed.

Two general types of change may be envisioned. The first is a controlled change, where the generator of the CDP stream can insert an explicit indication that service information has changed, or that a service has been added or dropped. The second is an uncontrolled change that could be caused by a switch from one CDP stream to another CDP stream. In the case of an unsupervised switch, there can be no controlled signaling of the change, yet the receiver of the CDP stream must be able to easily detect that a change has occurred. In the event that a CDP stream is switched, the switch could result in a stream that has an incomplete, wrong, or damaged caption descriptor that should be discarded (and not transmitted to a consumer DTV receiver). To provide the ability to detect stream switches, 16-bit sequence counts are included in the CDP header and footer. If the received sequence counts do not increment smoothly, a switch or error has occurred.

11.1.5 Interface data rates

The data rate required to convey a CDP stream is dependent on the frame rate of the stream and on the amount of service information included within each packet. The worst-case data rate would occur at a 60 Hz frame rate, with 16 services all fully described within each CDP packet. In this case, the CDP packet size would be 161 bytes, and require a transmission rate of 9,660 bytes per second, or 96,600 bps over a serial interface with 1 start bit and 1 stop bit. If the service information is limited to describing only 1 service per CDP, then the maximum data rate becomes 3,360 bytes per second, or 33.6 kbps over the serial interface. It is therefore practical to carry the CDP stream over an TIA/EIA-574 38.4 kbps serial interface, although in some cases it is necessary to limit the amount of service information included in each CDP packet.

11.2 CDP Detailed Specification

The CDP shall be as specified in Section 11.2.

11.2.1 General Construct

The general construct of the CDP shall be as defined in Table 23.

Syntax	Comment
cdp() {	Caption Distribution Packet (CDP)
cdp_header();	Required
time_code_section();	Optional
ccdata_section();	Optional
ccsvcinfo_section();	Optional
future_section_1();	If defined, place here
future_section_2();	If defined, place here
cdp_footer();	Required
}	

Table 23 CDP General Construct

The CDP shall contain one header section and one footer section. The CDP may contain one time code section, one cc data section, and one cc service information section. The CDP shall not contain more than one of any of these

sections. These sections shall be multiplexed in the order shown in Table 23. It is possible to extend the CDP to include additional sections. Any sections that are defined in the future shall be placed just prior to the cdp_footer. Any newly defined sections would begin with a unique identifier byte, and contain a length code. The syntax that a new section would follow is shown in Section 11.2.7. Equipment that receives the CDP shall skip over sections that begin with an unknown id byte, by means of the length code.

11.2.2 cdp_header

The CDP header is a required element, and shall be present in all CDP's. CDP header syntax shall be as indicated in Table 24. The length of the cdp_header is fixed at 6 bytes.

Syntax	Bits	Format	Comment
cdp_header() {			
cdp_identifier	16	uimsbf	0x9669
cdp_length	8	uimsbf	length of entire packet
cdp_frame_rate	4	uimsbf	frame rate of packets
Reserved	4	'1111'	
time_code_present	1	bit	'1' indicates time code section is included
ccdata_present	1	bit	'1' indicates cc data section is included
svcinfo_present	1	bit	'1' indicates svcinfo section is included
svc_info_start	1	bit	'1' indicates start of svc info data set
svc_info_change	1	bit	'1' indicates change in cc service information
svc_info_complete	1	bit	'1' indicates completion of svc info data set
caption_service_active	1	bit	'1' indicates caption service is active
Reserved	1	'1'	
cdp_hdr_sequence_cntr	16	uimsbf	
}			

Table 24 CDP Header Syntax

cdp_identifier – This is a 16-bit value set to 0x9669. All CDP packets begin with this value.

cdp_length – This 8-bit integer shall indicate the number of bytes of data in the entire CDP packet, from the first byte of the CDP_identifier, to the packet checksum, inclusive.

cdp_frame_rate – This field shall indicate the frame rate of the CDP stream. It shall be coded as indicated in Table 25. Also shown are the values of cc_count and the number of cc_data bytes that shall be included in each packet at each frame rate.

cdp_frame_rate	frame rate	cc_count	NTSC cc_data bytes	DTV cc_data bytes
0000	Forbidden			
0001	24000÷1001 (~23.976)	25	6	44
0010	24	25	6	44
0011	25	24	----	50
0100	30000÷1001 (~29.97)	20	4	36
0101	30	20	4	36
0110	50	12	----	25
0111	60000÷1001 (~59.94)	10	2	18
1000	60	10	2	18
...	Reserved			
1111	Reserved			

Table 25 CDP Frame Rate

NOTE--There is no practical difference between the pairs of frame rates which differ by 0.1%. Captions and pictures rendered at one rate may be played 0.1% fast or slow, with no impact on presentation, as the number of bytes per frame does not change.

time_code_present – This bit shall be set to '1' for CDP packets which include the time code section. Otherwise this bit shall be set to '0'.

cc_data_present – This bit shall be set to '1' for CDP packets which include the cc data section. Otherwise this bit shall be set to '0'.

svcinfo_present – This bit shall be set to '1' for CDP packets which include the service information section. Otherwise this bit shall be set to '0'.

svc_info_start – This bit shall be set to '1' to indicate that the current packet begins a complete set of service information. Otherwise this bit shall be set to '0'. This bit shall be set to '0' if this CDP packet does not contain a service information section. This bit is duplicated in the cc service information section. The value of this bit shall not be different from the value of the svc_info_start bit in the cc service information section.

svc_info_change – This bit shall be set to '1' during the packet which begins a complete set of service information to indicate that the service information in the following set of information has changed from the previously delivered set of information. Otherwise this bit shall be set to '0'. This bit shall be set to '0' if this CDP packet does not contain a service information section. This bit is duplicated in the cc service information section. The value of this bit shall not be different from the value of the svc_info_change bit in the cc service information section.

svc_info_complete – This bit shall be set to '1' to indicate that the current packet concludes a full set of service information. Otherwise this bit shall be set to '0'. This bit shall be set to '0' if this CDP packet does not contain a service information section. This bit is duplicated in the cc service information section. The value of this bit shall not be different from the value of the svc_info_complete bit in the cc service information section.

caption_active – This bit shall be set to '1' to indicate that the CDP stream is conveying an active caption service. This bit shall be set to '0' in the case that the CDP stream is not conveying an active caption service.

cdp_hdr_sequence_cntr – This is an unsigned 16-bit integer which shall be set to a value of 1 plus the value of CDP_hdr_sequence_cntr in the previous CDP. The value of this counter shall wrap from 65535 to 0. For the first CDP in a sequence of CDPs, dccp_hdr_sequence_cntr may be set to any 16-bit value.

11.2.3 time_code_section

The time code section is optional in a CDP. Time code syntax is indicated in Table 26. This section shall be composed of a section id byte and 4 bytes of time code information. The length of the time code section shall be 5 bytes. Inclusion of this section may help assure that synchronization between captions and pictures is maintained throughout the distribution chain and into the final emission transport stream.

Syntax	Bits	Format	Comment
time_code_section() {			
time_code_section_id	8	uimsbf	0x71
Reserved	2	'11'	
tc_10hrs	2	uimsbf	Tens of hours
tc_1hrs	4	uimsbf	Units of hours
Reserved	1	'1'	
tc_10min	3	uimsbf	Tens of minutes
tc_1min	4	uimsbf	Units of minutes
tc_field_flag	1	uimsbf	see text
tc_10sec	3	uimsbf	Tens of seconds
tc_1sec	4	uimsbf	Units of seconds
Reserved	1	'1'	
drop_frame_flag	1	uimsbf	Drop frame flag
tc_10fr	3	uimsbf	Tens of frames
tc_1fr	4	uimsbf	Units of frames
}			

Table 26 CDP Time Code Section Syntax

time_code_section_id – This 8-bit field shall have the value of 0x71.

tc_field_flag – For interlaced pictures, the value of this flag shall be '0' for interlace field 1, and shall be '1' for interlace field 2. In the case of frame rates equal to or greater than 50 Hz, the frame count shall be interpreted as follows. The frame count shall be doubled, and the tc_field_flag shall be interpreted as an adder to the indicated frame count. The frame count shall be interpreted as (2 * frame + flag). I.e. the frame:flag sequence shall be 0:0, 0:1, 1:0, 1:1, 2:0, 2:1, etc., and this frame:flag sequence shall be interpreted as progressive frame counts 0, 1, 2, 3, 4, 5, etc.

drop_frame_flag – This flag shall be set to '1' when the time code count is being drop-frame compensated.. When the count is not drop-frame compensated, this flag bit shall be set to '0'.

11.2.4 ccdata_section

The ccdata section should normally be present. If present, the ccdata section syntax shall be as indicated in Table 27.

This section need not be included if a CDP stream is intended to carry only caption service information to a PMT or PSIP generator. However, a CDP stream typically conveys both cc data and cc service information in parallel to both the emission encoder and to the PMT and PSIP generators, with each device extracting and using the appropriate information.

This section shall be composed of a section id byte, a count value cc_count, and cc_count groups of 3 bytes. The total length of this section is 2 + 3 * cc_count bytes. The value of cc_count shall be dependent on the frame rate that is indicated in the CDP_header.

The actual caption data is carried in the cc_data_1 and cc_data_2 fields. The value of cc_count is found in table 10-2, and provides sufficient space to carry both NTSC and DTV caption data. The DTV caption data may represent up

to 16 caption services. If either the NTSC or DTV caption data is not present, the space is still allocated and filled with null (0x0) values. The NTSC caption data shall come first in this section, followed by the DTV caption data.

Syntax	Bits	Format	Comment
ccdata_section() {			
ccdata_id	8	0x72	Indicates ccdata section
marker_bits	3	'111'	
cc_count	5	uimsbf	number of cc constructs in section
for (i = 0 ; i < cc_count ; i++) {			
marker_bits	5	'1111 1'	
cc_valid	1	bslbf	as defined in 4.4.1
cc_type	2	bslbf	as defined in 4.4.1
cc_data_1	8	bslbf	
cc_data_2	8	bslbf	
}			
}			

Table 27 CDP CC Data Section Syntax

ccdata_id – This 8-bit field shall have the value 0x72.

cc_count – This 5-bit field shall indicate the number of cc_[xxx] data byte triplets carried in this section, and shall have the value appropriate to the frame rate as indicated in Table 25.

cc_valid – This bit shall be as defined in section 4.4.1. This bit indicates whether the following two bytes of cc_data contain valid captioning data or zero padding.

cc_type – This 2-bit field shall be as defined in section 4.4.1. This field indicates whether the following two bytes of data represent NTSC field 1 or 2 capions, or DVC channel packet data or DTVCC channel packet start.

cc_data_1 – This byte shall be as defined in section 4.4.3.

cc_data_2 – This byte shall be as defined in section 4.4.3.

11.2.5 ccsvcinfo_section

The ccsvcinfo section carries information for the Caption Service Descriptor. This section shall be composed of a section id byte, indication of controlled changes in the service information, an indication of the number of services that are described in the current packet, and caption service information. The ccsvcinfo_section syntax shall be as described in Table 28.

The complete set of caption service information may describe 1 to 16 different caption services. A complete set of service information may be included in the current packet, or may be distributed over a number of packets. The total length of this section is 2 + 7 * svc_count bytes.

Syntax	Bits	Format	Comment
ccsvcinfo_section() {			
ccsvcinfo_id	8	0x73	Indicates ccsvcinfo section
marker_bit	1	'1'	
svc_info_start	1	bit	
svc_info_change	1	bit	
svc_info_complete	1	bit	
svc_count	4	uimsbf	number of svc constructs in section
for (i = 0 ; i < svc_count ; i++) {			
reserved	3	'111'	
caption_service_number	5	uimsbf	
svc_data_byte_1	8	bslbf	
svc_data_byte_2	8	bslbf	
svc_data_byte_3	8	bslbf	
svc_data_byte_4	8	bslbf	
svc_data_byte_5	8	bslbf	
svc_data_byte_6	8	bslbf	
}			
}			

Table 28 CC Service Information Syntax

svc_info_start – This bit shall be set to '1' to indicate that the current packet begins a complete set of service information. Otherwise this bit shall be set to '0'. This bit shall be duplicated in the CDP header section. The value of this bit shall not be different from the value of the svc_info_start bit in the CDP header section.

svc_info_change – This bit shall be set to '1' during the packet which begins a complete set of service information to indicate that the service information in the following set of information has changed from the previously delivered set of information. Otherwise, this bit shall be set to '0'. This bit is duplicated in the CDP header section. The value of this bit shall not be different from the value of the svc_info_change bit in the CDP header section.

svc_info_complete – This bit shall be set to '1' to indicate that the current packet concludes a full set of service information. Otherwise this bit shall be set to '0'. This bit is duplicated in the CDP header section. The value of this bit shall not be different from the value of the svc_info_complete bit in the CDP header section.

> NOTE--If a single packet contains a complete set of service information, then both the svc_info_start and svc_info_end bits would be set to '1'.

svc_count – This 4-bit field shall be set to a value equal to the number of services which have service information included in this service information section.

caption_service_number – This 5-bit field carries the caption service number for the service described by the following 6 service data bytes. This field shall have a value of 0x00 when the service data applies to the line 21 (EIA-608-A) service, and shall have a value (between 0x01 – 0x10 inclusive) that matches the caption_service_number contained within svc_data_byte_4 when the service data applies to one of the DTVCC services. This field shall not have a value between 0x11 and 0x1f (inclusive).

svc_data_byte_*n* – These 6 bytes shall carry the caption service data for one service, encoded as described by the caption service descriptor loop in ATSC A/65, section 6.7.3, table 6.17.

11.2.5.1 Service information signalling

The svc_info start, change, and end bits allow the receiver of a CDP stream to build up a complete set of cc service information from multiple packets, and to follow changes in the captioning services which are intentionally introduced by the source of the CDP stream. These changes may include beginning and ending of one or more captioning services, or an alteration in a particular service.

When a service is terminated, the svc_info_change bit shall be set to '1' during the first packet containing the next full set of service information. The following full sets of service information shall not contain any service information for the terminated service. The caption service number of the terminated service shall not appear in any of the svc info sections.

When a service is changed, the svc_info_change bit shall be set to '1' during the first packet containing the next full set of service information. The following full sets of service information shall contain service information representing the new service information for the changed service.

If a CDP stream is switched, there should be a discontinuity in the sequence counters. If the switch occurs between packets, the discontinuity will occur between the previous footer value and the following header value. If the switch occurs during a packet, the balance of the new packet may not be received correctly and there will not be a correct sequence number or checksum in the footer. If a receiver detects that a CDP stream switch has occurred, the receiver should assume all service information has changed and take the next full set of service information as current.

> NOTE--In this case, the svc_info_change bit will not signal a change of service information. Uncontrolled switches of CDP streams may cause momentary glitches in the display of captions on receivers.

11.2.6 cdp_footer

The CDP footer shall be present in all CDP packets. The CDP footer syntax shall be as defined in Table 29. This section contains a section id byte, a sequence counter value, and a checksum. The sequence counter provides good detection ability as to whether a switch occurred during the transmission of the current CDP packet. The check sum provides a simple means of detecting most transmission errors. The total length of this section is 4 bytes.

Syntax	Bits	Format	Comment
cdp_footer() {			
cdp_footer_id	8	0x74	Indicates CDP footer section
cdp_ftr_sequence_cntr	16	uimsbf	
packet_checksum	8	uimsbf	
}			

Table 29 CDP Footer Syntax

cdp_ftr_sequence_cntr – This 16-bit unsigned integer shall be set to the same value as the cdp_hdr_sequence_cntr. Receivers may use the values of cdp_hdr_sequence_cntr and cdp_ftr_sequence_cntr to detect that the entire packet has been received.

packet_checksum – This 8-bit field shall contain the 8-bit value necessary to make the arithmetic sum of the entire packet (first byte of cdp_identifier to packet_checksum, inclusive) modulo 256 equal zero.

11.2.7 future_section()

It is possible to define new sections to be included in the CDP. Any newly defined sections shall follow the syntax defined in this clause and in Table 30. All equipment that can receive the CDP shall be capable of ignoring these new sections. The length value is provided so that decoders will know how many bytes of data to skip.

Syntax	Bits	Format	Comment
future_section() {			
future_section_id	8	uimsbf	Value in range 0x75-0xEF
Length	8	uimsbf	Number of bytes of data
for (i = 0 ; i < length ; i++) {			
new_data_byte(i)	8		New data content
}			
}			

Table 30 future_section syntax

future_section_id – Sections defined in the future shall be specified to have a section id value in the range 0x75 to 0xEF, inclusive. Decoders shall be designed to skip over sections that have id values that are not understood.

11.3 Serial Interface

This section defines one specific serial interface which may convey a CDP stream.

11.3.1 Physical Interface

The physical interface for a CDP stream shall be an TIA/EIA-574 interface. The source of the CDP stream shall be DTE with a 9-pin "D" male connector, and the receiver of the CDP stream shall be DCE with a 9-pin "D" female connector.

Table 31 and Table 32 detail the pin connections for the CDP serial interface. The terminology under the "Signal Name" column reflects that used in TIA/EIA-574. The source of the CDP stream shall follow the settings shown in Table 11-7. The receiver of the CDP stream shall follow the settings shown in Table 11-8. The minimum implementation of this serial interface only requires two wires: serial data and ground. In the case where the other 7 lines are connected to drivers, the state of those lines is specified.

Pin	In or Out	Signal Name	Setting	Comments
1	Input	Receive Line Signal Detect	NA	NC or ignored
2	Input	Receive Data	NA	NC or ignored
3	Output	Transmit Data	Active Data	CDP data stream
4	Output	DTE Ready	NC or "ON"	If connected, set "ON"
5	---	Common	GND	Signal common
6	Input	DCE Ready	NA	NC or ignored
7	Output	RTS	NC or "ON"	If connected, set "ON"
8	Input	CTS	NA	NC or ignored
9	Input	Ring Indicator	NA	NC or ignored

Table 31 CDP Source Connector Pinout (DTE Male)

Pin	In or Out	Signal Name	Setting	Comments
1	Output	Receive Line Signal Detect	NC or "ON"	If connected, set "ON"
2	Output	Receive Data	NC or "ONE"	If connected, set to "ONE" (Mark)
3	Input	Transmit Data	Active Data	DCCCP data stream
4	Input	DTE Ready	NA	NC or ignored
5	---	Common	GND	Signal common
6	Output	DCE Ready	NC or "ON"	If connected, set to "ON"
7	Input	RTS	NA	NC or ignored
8	Output	CTS	NC or "ON"	If connected, set to "ON"
9	Output	Ring Indicator	NC or "OFF"	If connected, set to "OFF"

Table 32 CDP Receiver Connector Pinout (DCE Female)

NOTE--NC means not connected; NA means not applicable.

The serial interface shall support operation with the parameters indicated in Table 33. The recommended baud rate is 38,400, but, if necessary, the interface may also be operated at 57,600 or 115,200 baud.

Parameter	Setting
Baud Rate	38,400 (default); 57,600 (optional); 115,200 (optional
Data Bits	8
Parity	None
Stop Bits	1
Start Bits	1

Table 33 TIA/EIA-574 Interface Parameters for CDP Stream

11.3.2 Operation of the CDP Serial Interface

This section describes the typical application of the CDP serial interface. Other applications are not precluded.

In a typical application, captioning intentions are captured in a high level representation and then rendered into EIA-708 captioning packets. SMPTE time code may be employed to provide means for synchronizing the captioning intentions to the picture. With knowledge of the picture frame rate and time code, the EIA-708 captioning packets may be formed into CDP packets, where one CDP packet corresponds to each picture frame. During real-time streaming of pictures over a video interface and the corresponding CDP packets over the CDP serial interface, the CDP packet for picture frame n should be presented to the serial interface during the time window between the beginning of picture frame n and the beginning of picture frame n+1. In an ideal application, both the CDP packets and the individual pictures will have timecodes, and the MPEG-2 picture encoder will rely on these time codes to establish synchronization between the MPEG-2 encoded pictures and the captioning data included in the user data space of those coded pictures.

When the CDP is conveyed by this serial interface, the CDP shall be preceded by four null bytes (0x00). These null bytes, plus the cdp_identifier (0x9669), form a unique 48-bit sync code that allows the serial receiver to synchronize to the CDP stream.

NOTE—These null bytes are not considered part of the CDP, and are not required when the CDP is carried by other interfaces.

BIBLIOGRAPHY

ATSC A/54, Guide to the Use of the ATSC Digital Television Standard (1995)

- Advanced Television Systems Committee (ATSC), 1750 K Street N.W., Suite 1200, Washington, DC 20006; Phone 202-828-3130; Fax 202-828-3131; Internet http://www.atsc.org/stan&rps.html

Annex A (Informative)

Figure 22 illustrates one example of a decoder implementation for processing the PMT, EIT and User Data in the DTVCC Transport Stream. It shows the separate paths of the Service Descriptors in the PMT and EIT and the service data in the Picture User Data bits.

NOTE--Different block diagrams may exist for other implementations.

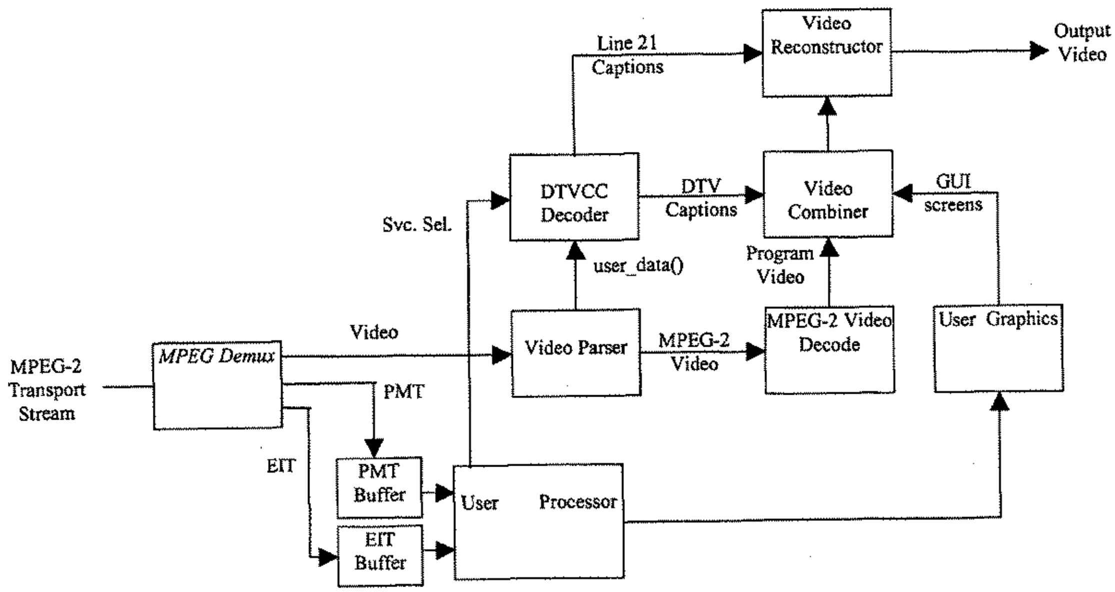

Figure 22 DTVCC Transport Stream Decoder

EIA Document Improvement Proposal

If in the review or use of this document, a potential change is made evident for safety, health or technical reasons, please fill in the appropriate information below and mail or FAX to:

Electronic Industries Alliance
Engineering Department – Publications Office
2500 Wilson Blvd.
Arlington, VA 22201
FAX: (703) 907-7501

Document No.	Document Title:
Submitter's Name:	Telephone No.: FAX No.: e-mail:
Address:	
Urgency of Change: Immediate: ☐	At next revision: ☐
Problem Area: a. Clause Number and/or Drawing: b. Recommended Changes: c. Reason/Rationale for Recommendation:	
Additional Remarks:	
Signature:	Date:
FOR EIA USE ONLY Responsible Committee: Chairman: Date comments forwarded to Committee Chairman:	

Electronic Industries Alliance

CEA Standard

Digital Television (DTV) Closed Captioning

CEA-708-C

July 2006

Consumer Electronics Association

www.CE.org

NOTICE

CEA Standards, Bulletins and other technical publications are designed to serve the public interest through eliminating misunderstandings between manufacturers and purchasers, facilitating interchangeability and improvement of products, and assisting the purchaser in selecting and obtaining with minimum delay the proper product for his particular need. Existence of such Standards, Bulletins and other technical publications shall not in any respect preclude any member or nonmember of CEA from manufacturing or selling products not conforming to such Standards, Bulletins or other technical publications, nor shall the existence of such Standards, Bulletins and other technical publications preclude their voluntary use by those other than CEA members, whether the standard is to be used either domestically or internationally.

Standards, Bulletins and other technical publications are adopted by CEA in accordance with the American National Standards Institute (ANSI) patent policy. By such action, CEA does not assume any liability to any patent owner, nor does it assume any obligation whatever to parties adopting the Standard, Bulletin or other technical publication.

This CEA Standard is considered to have International Standardization implication, but the International Electrotechnical Commission activity has not progressed to the point where a valid comparison between the CEA Standard and the IEC document can be made.

This Standard does not purport to address all safety problems associated with its use or all applicable regulatory requirements. It is the responsibility of the user of this Standard to establish appropriate safety and health practices and to determine the applicability of regulatory limitations before its use.

(Formulated under the cognizance of the CEA's **R4.3 Television Data Systems Subcommittee**.)

Published by

©CONSUMER ELECTRONICS ASSOCIATION 2006
Technology & Standards Department
2500 Wilson Boulevard
Arlington, VA 22201

PRICE: Please call Global Engineering Documents, USA and Canada (1-800-854-7179)
International (303-397-7956), or
http://global.ihs.com

Contents

Figures

Tables

FOREWORD

CEA-708-C defines a method for coding text with associated parameters to control its display. CEA-708-C is the standard for Closed Captioning in Digital Television (DTV) technology. Predecessors of CEA-708-C was originally developed under the auspices of the Consumer Electronics Association (CEA) Technology & Standards R4.3 Television Data Systems Subcommittee in parallel with the U.S. Advanced Television Systems Committee's (ATSC) and the Advanced Television Grand Alliance's definition, design, and development of the audio, video and ancillary data processing standard for Advanced Television. The DTV standard developed by the Grand Alliance and other industry members is documented in ATSC A/53E, and the informative document, ATSC A/54A.

CEA-708-C supersedes EIA-708-B and CEA-708-B, CEA-CEB10-A, and CEA-CEB8.

DIGITAL TELEVISION (DTV) CLOSED CAPTIONING

1 Scope

CEA-708-C defines DTV Closed Captioning (DTVCC) and provides specifications and guidelines for caption service providers, distributors of television signals, decoder and encoder manufacturers, DTV receiver manufacturers, and DTV signal processing equipment manufacturers. CEA-708-C may also be useful in other systems. CEA-708-C includes the following:

a) a description of the transport method of DTVCC data in the DTV signal
b) a specification for processing DTVCC information
c) a list of minimum implementation recommendations for DTVCC receiver manufacturers
d) a set of recommended practices for DTV encoder and decoder manufacturers

The use of the term DTV throughout is intended to include, and apply to, High Definition Television (HDTV) and Standard Definition Television (SDTV).

1.1 Overview

DTVCC is a migration of the closed-captioning concepts and capabilities developed in the 1970's for NTSC television video signals to the digital television environment defined by the ATV (Advanced Television) Grand Alliance and standardized by ATSC. This new television environment provides for larger screens and higher screen resolutions, as well as higher data rates for transmission of closed-captioning data.

National Television Systems Committee II (NTSC) Closed Captioning (CC) consists of an analog waveform inserted on line 21, field 1 and possibly field 2, of the NTSC Vertical Blanking Interval (VBI). That waveform provides a transport channel which can deliver 2 bytes of data on every field of video. This translates to a nominal 120 bytes per second (Bps), or a nominal 960 bits per second (bps).

In contrast, DTV Closed Captioning is transported as a logical data channel in the DTV digital bitstream. DTV-specific closed captioning is allocated 9600 bps for each program. This increased capacity opens the possibility for simultaneous transmission of captions in multiple languages and with multiple reading levels, as well as the transport of an entire CEA-608-C datastream.

The DTV standard also accommodates a variety of increased horizontal and vertical resolutions (e.g., 704x480, 1280x720 and 1920x1080), versus the single 525 vertical scan line format for NTSC. These added resolutions provide for more defined representations of character fonts and other on-screen objects.

The heart of any DTVCC caption display is the caption "window," which is similar to the *window* concept found in many computer Graphical User Interfaces (GUIs). Windows are placed within the DTV screen, and caption text is placed within windows. Windows and text have a variety of color, size and other attributes.

CEA-708-C describes the above issues in a reverse-hierarchical (i.e., low-to-high level) fashion. It follows an "Open Systems Interconnect (OSI) Reference Model"-type protocol stack for layered protocols. DTVCC consists of 5 protocol layers: the Transport Layer, the Packet Layer, the Service Layer, the Coding Layer, and the Interpretation Layer. The discussion of the first 2 layers is a detailed presentation of data organization issues. The discussion of the last 2 layers provides a more informative presentation of the unique aspects of closed captioning. Some readers may wish to start with these last 2 layers first, beginning in Section 7.

1.2 Notation

Designers should interpret CEA-708-C syntax and values based on notational conventions taken from the referenced Motion Picture Experts Group (MPEG), ATSC and CEA standards. Numbering and counting loops generally begin from zero unless otherwise specified. Some specific cases and examples are provided in the following sections to assist in clarifying notations used in CEA-708-C.

1.2.1 Binary Notation
The following notational convention is used in CEA-708-C to describe binary numbers: 'b111'. The number of digits following the b may vary and, as do bit strings, may have a space every 4 digits.

1.2.2 Hex Notation
CEA-708-C uses a "C-like" notational convention to describe hexadecimal numbers, e.g. "0x36".

1.2.3 Equals and Assignment
The assignment operator "=" and the Equal To relational operator "==" are freely mixed in CEA-708-C. As in PASCAL or BASIC programming languages, the single equal symbol can be used to mean, "is equal to", in addition to "is assigned the value". The text may also use the equal symbol to mean, "evaluates to the value of", as in: "864/75 = 11.52". The interpretation of the single equal symbol within CEA-708-C depends on context and the most logical reading of the text.

1.2.4 Bitstream Syntax Notation
Where fields have the individual bits numbered, the lowest numbered bit in a multi-bit numbered value is the least significant bit.

In some syntactical structures mnemonics are used to define the order of bits in that field. The mnemonic "bslbf" is defined to be bit string, left bit first, where left-to-right is the order in which bit strings are written.

> NOTE—Bit strings are written as a string of 1s and 0s within single quote marks, e.g. '1000 0001'. Blanks within a bit string are typically every four bits for ease of reading and have no significance.

1.2.5 Acronyms & Mnemonics
AES	Audio Engineering Society
ANSI	American National Standards Institute
ASCII	American Standard Code for Information Exchange
ATSC	Advanced Television Systems committee
ATV	Advanced Television
bg	Background
bps	Bits per Second
Bps	Bytes per Second
bs	Block Size
BS	Back Space
bslbf	Bit String Left Bit First
CC	Closed caption
cc	Column Count
CEA	Consumer Electronics Association
CEB	Consumer Electronics Bulletin
cl	Column Lock
CLW	Clear Window
CR	Carriage Return
DFx	Define Window
DLC	Delay Cancel
DLW	Delete Windows
DLY	Delay
DSW	Display Windows
DTV	Digital Television
DTVCC	Digital Television Closed Caption
EIA	Electronic Industries Alliance
EIT	Event Information table
ETX	End of Text
fg	Foreground
FIFO	First In First Out
fps	Frames per Second
FF	Form Feed
FG	Foreground
GOP	Group of Pictures

GUI Graphical User Interface
HCR Horizontal Carriage Return
HDW Hide Windows
HDTV High Definition Television
ID Identification
IEC International Electrotechnical Commission
ISO International Standards Organization
LCD Liquid Crystal Display
MPEG Motion Picture Experts Group
NBS Non-Breaking Space
NBTSP Non-Breaking Transparent Space
NTSC National Television Systems Committee
NUL Null
OSI Open Systems Interconnection
PMT Program Map table
PSIP Program and System Information Protocol
rc Row Count
RF Radio Frequency
RGB Red Green Blue
RST Reset
SDTV Standard Definition Television
SMPTE Society of Motion Picture and Television Engineers
SPA Space
SPC Set Pen Color
SPL Set Pen Location
STB Set Top Box
SWA Set Window Attributes
TGW Toggle Windows
TSP Transparent Space
uimsbf Unsigned Integer Most Significant Bit First
VBV Video Buffering Verifier
XDS eXtended Data Services

2 Normative References

The following references contain provisions, which, through reference in this text, constitute normative provisions of this standard. At the time of publication, the edition indicated was valid. All standards are subject to revision, and parties to agreements based on this standard are encouraged to investigate the possibility of applying the most recent edition of the standard indicated below.

2.1 Normative References

2.1.1 Normative Reference List

ANSI INCITS 4, Information Systems—Coded Character Sets—7 Bit American National Standard Code for Information Interchange (7-BIT ASCII), 2002

ATSC A/53E, ATSC Digital Television Standard, December 27, 2005

ATSC A/65C, Program and System Information Protocol for Terrestrial Broadcast and Cable (January 2, 2005)

CEA-608-C, Line 21 Data Service, August, 2005

ISO/IEC 2022, Information technology - Character code structure and extension techniques (4/15/1999)

ISO/IEC 8859-1, Information technology - 8-bit single-byte coded graphic character sets—Part 1: Latin alphabet No. 1 (4/15/1998)

ISO/IEC IS 13818-1:2000 (E), Information technology - Generic coding of moving pictures and associated audio information: systems (2000)

ISO/IEC IS 13818-2:2000 (E), International Standard, Information technology—Generic coding of moving pictures and associated audio information: video (2000)

SMPTE RP 218-2002, Specifications for Safe Action and Safe Title Areas for Television Systems

SMPTE Standard 12M, Television, Audio and File—Time and Control Code (1999)

2.1.2 Normative Reference Acquisition
ANSI or CEA Standards:

- Global Engineering Documents, World Headquarters, 15 Inverness Way East, Englewood, CO USA 80112-5776; Phone 800-854-7179; Fax 303-397-2740; Internet http://global.ihs.com; Email global@ihs.com

ATSC Standards:

- Advanced Television Systems Committee (ATSC), 1750 K Street N.W., Suite 1200, Washington, DC 20006; Phone 202-872-9160; Fax 202-872-9161;; Internet http://www.atsc.org A

ISO/IEC Standards:

- Global Engineering Documents, World Headquarters, 15 Inverness Way East, Englewood, CO USA 80112-5776; Phone 800-854-7179; Fax 303-397-2740; Internet http://global.ihs.com; Email global@ihs.com
- IEC Central Office, 3, rue de Varembe, PO Box 131, CH-1211 Geneva 20, Switzerland; Phone +41 22 919 02 11; Fax +41 22 919 03 00; Internet http://www.iec.ch; Email pubinfor@iec.ch
- ISO Central Secretariat, 1, rue de Varembe, Case postale 56, CH-1211 Geneve 20, Switzerland; Phone +41 22 749 01 11; Fax +41 22 749 01 55; Internet http://www.iso.ch; Email mbinfo@iso.ch

SMPTE Standards:

- Society of Motion Picture & Television Engineers, 3 Barker Ave., White Plains, NY 10601 USA Phone: 914-761-1100 Fax: 914-761-3115, Email: eng@smpte.org; Web: http://www.smpte.org

2.2 Informative References

2.2.1 Informative Reference List
AES3-2003, AES Recommended practice for digital audio engineering -- Serial transmission format for two-channel linearly represented digital audio data

ATSC A/54A, Guide to the Use of the ATSC Digital Television Standard (December 4, 2003)

ATSC A/76, Programming Metadata Communication Protocol (November 10, 2004)

ISO 639-2:1998, Codes for the representation of names of languages—Part 2: Alpha-3 code

SMPTE EG 43-2004, System Implementation of CEA-708-B and CEA-608-B Closed Captioning

SMPTE 259M-1997, Television – 10-Bit 4:2:2 Component and 4fsc Composite Digital Signals – Serial Digital Interface

SMPTE 292M-1998, Television – Bit-Serial Digital Interface for High-Definition Television Systems

TIA-232-F, Interface between Data Terminal Equipment and Data Circuit-Terminating Equipment Employing Serial Binary Data Interchange (October 11, 2002)

2.2.2 Informative Reference Acquisition
AES Documents:

- Audio Engineering Society, International Headquarters, 60 East 42nd Street, Room 2520, New York, New York, 10165-2520, USA, Tel: +1 212 661 8528, Fax: +1 212 682 0477, Internet: http://www.aes.org

ATSC Standards:

- Advanced Television Systems Committee (ATSC), 1750 K Street N.W., Suite 1200, Washington, DC 20006; Phone Phone 202-828-3130; Fax 202-828-3131; Internet http://www.atsc.org

SMPTE Standards:

- Society of Motion Picture & Television Engineers, 595 W. Hartsdale Ave., White Plains, NY 10607-1824 USA Phone: 914-761-1100 Fax: 914-761-3115, Email: eng@smpte.org; Web: http://www.smpte.org

TIA Standards:

- Global Engineering Documents, World Headquarters, 15 Inverness Way East, Englewood, CO USA 80112-5776; Phone 800-854-7179; Fax 303-397-2740; Internet http://global.ihs.com; Email: global@ihs.com

ISO Standards:

- Global Engineering Documents, World Headquarters, 15 Inverness Way East, Englewood, CO USA 80112-5776; Phone 800-854-7179; Fax 303-397-2740; Internet http://global.ihs.com; Email global@ihs.com
- ISO Central Secretariat Office, 1, rue de Varembe, Case postale 56, CH-1211 Geneve 20, Switzerland; Phone +41 22 749 01 11; Fax +41 22 733 34 30; Internet http://www.iso.ch; Email central@iso.ch

3 Caption Channel Layered Protocol

A formal data communications channel protocol has been established for the DTVCC caption data channel. This formalization provides a framework for describing the caption communications hierarchy. Grouping the structures, concepts, and features of this environment into the following hierarchical layers aids in the understanding of the organizational aspects of the DTVCC system.

There are 5 layers in the DTVCC data framework: DTVCC Transport Layer, DTVCC Packet Layer, DTVCC Service Layer, DTVCC Coding Layer, and DTVCC Interpretation Layer (see Figure 1). Similar to the OSI Reference Model, each of these layers provides particular services in the receiver, as shown in Table 1.

Services in a Receiver	CEA-708-C Section	DTVCC Protocol Model	CEA-608-C Datastream
• Specification of Window appearance and content • Placement of Windows over Video • Synchronization of Command Interpretation to the Display of video • Reset of Caption Service	8	DTVCC Interpretation Layer	These layers are defined in CEA-608-C.
• Parsing of Syntax	7	DTVCC Coding Layer	
• Demultiplexing of Caption Services	6	DTVCC Service Layer	
• Detection of Synchronization Loss • Resynchronization of Data Stream • Reset of Caption Decoder	5	DTVCC Packet Layer	
• Extraction of closed caption data from parent transport system, e.g. MPEG user data • Demultiplexing of CEA-608-C datastream and DTVCC caption streams	4	DTVCC Transport Layer	

Table 1 DTVCC Protocol Stack

The DTVCC Transport Layer is where DTVCC data leave the DTV Video subsystem and are introduced to the DTV Closed-Caption decoder. Within the DTVCC decoder, DTVCC data are further processed up through the remaining layers of the DTVCC Protocol Model. Both the CEA-608-C datastream and DTVCC data are carried in this layer.

> NOTE—The DTVCC Transport layer should not be confused with the main system transport layer. For example, in MPEG-2 ATSC systems, the MPEG-2 transport layer is defined in ISO 13818-1, and this transport layer carries video, audio, data and control streams. DTVCC bits are encoded within the video signal conforming with ISO 13818-2. These bits enter the DTVCC decoder from the video stream's user data structure.

CEA-708-C specifies a method of placing a CEA-608-C datastream, embedded in the CEA-708-C signal, into the correct bit positions and fields in an NTSC signal, but CEA-708-C does not specify any application or presentation level processes for CEA-608-C datastreams. Further processing of CEA-608-C datastreams is specified in CEA-608-C.

The DTVCC Packet Layer marks where DTVCC data enter the DTVCC decoder. This is a protocol data reassembly layer that buffers incremental bitstream data into a byte-aligned, multi-byte packet. This layer provides resynchronization for decoding the DTVCC stream, and is not dependent on MPEG-2 user data structures.

Processing of the Caption Channel packet data begins in the DTVCC Service Layer. Caption Channel packets are broken up into the encapsulated sub-blocks of data to be routed to one or more separate caption service processing routines within the decoder.

Using information in the DTVCC service directory, such as that sent in the Event Information Table (EIT) or Program Map Table (PMT) (see ATSC A/53E and ATSC A/65C), TV viewers may choose to view the processed data for one or more DTVCC services at a time. For example, a caption channel may contain an English language service and a Spanish language service.

The DTVCC Coding Layer breaks out the individual caption commands and caption text sequences from the service data blocks.

The DTVCC Interpretation Layer processes the caption elements presented by the DTVCC Coding Layer.

The DTVCC Interpretation Layer also provides for delaying the interpretation of the command stream and reset of a specific Caption Service.

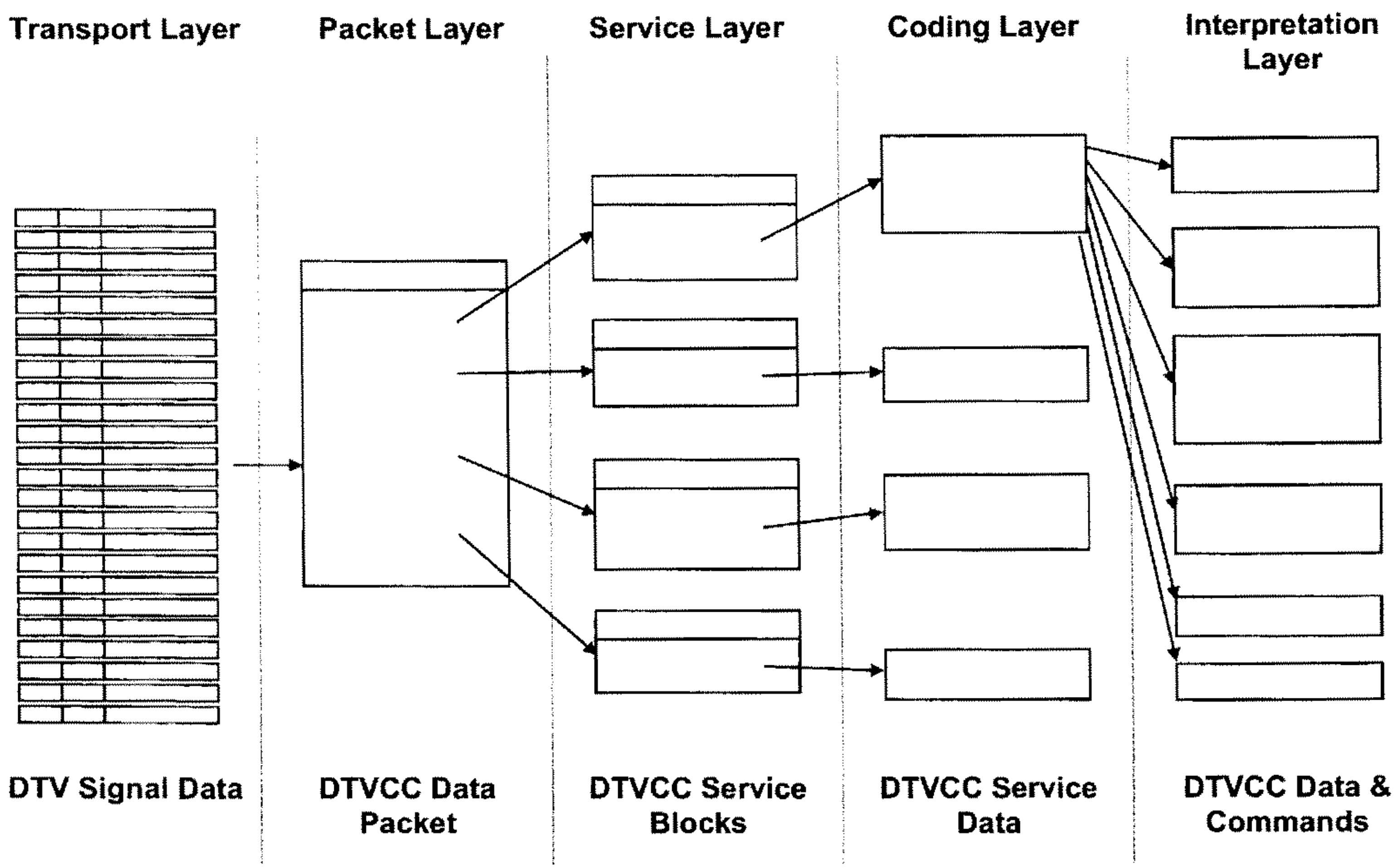

Figure 1 DTV Closed-Captioning Protocol Model

4 DTVCC Transport Layer

Section 4 describes the transport layer for closed caption information within the DTV Bitstream, carriage of the CEA-608-C datastream and DTVCC Caption Channel over the DTVCC Transport Channel, and carriage and semantic requirements of the DTVCC Service Directory information (via the **caption service descriptor()**) in the PMT and EIT.

The transport of the Caption Channel is defined in ATSC A/53E as an extension of ISO/IEC 13818-2. ATSC A/53E Table A7 describes the syntax for transport of closed caption data, clarified by ATSC A/53E Annex A Table A8. CEA-708-C defines the semantics of those syntax elements.

The DTVCC Transport Layer consists of the mechanisms for transporting caption data from the encoder at the caption-encoding head-end to the decoding hardware in the TV receiver. See Annex A. DTVCC related data, when present, is included in three separate portions of the DTV Bitstream:

a) Video User Data Bits, **user_data()**
b) PMT **caption_service_descriptor()**
c) EIT **caption_service_descriptor()**.

DTVCC Service Data (caption text, window commands, etc) are carried as MPEG-2 Picture User Data, and the DTVCC Caption Channel Service Directory is carried as descriptor information in the Program Map Table (PMT) and, when present, the Event Information Table (EIT). The EIT and the **caption_service_descriptor()** are defined in ATSC A/65C.

The DTV video bitstream, the PMT and the EIT are multiplexed with the other audio, data, control and synchronization bitstreams comprising the DTV system signal as depicted in Figure 2.

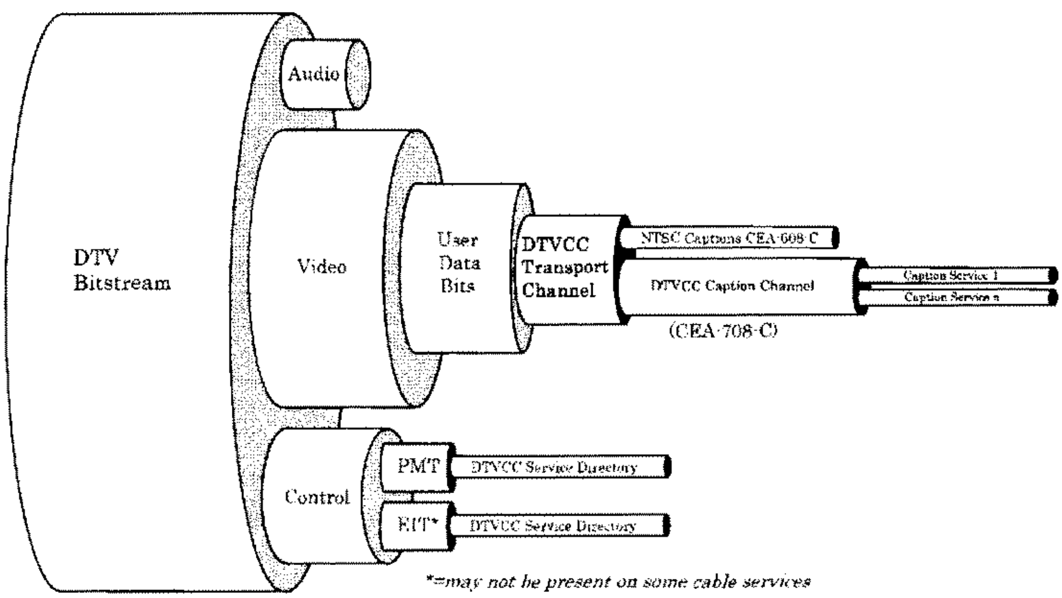

Figure 2 DTVCC Caption Data in the DTV Bitstream

4.1 9,600 Bits Per Second DTVCC Transport Channel

The DTVCC design allows for caption data to be transmitted at various data rates within a DTV video signal or any similar MPEG-2 variation. However, for DTV and DTVCC specific (i.e., ATSC A/53E) caption

encoding, the DTV Closed-Captioning Transport Channel is a continuous 9600 bits per second (bps) stream allocated from the DTV video signal capacity.

NOTE—This value of 9600 bps is divided by 1.001 when the **frame_rate_code** is 1, 4, or 7.

In order to provide this continuous stream and to ensure that specific caption data will reach receivers as intended (in relationship to the audio and video), this data channel is allocated on a frame-by-frame basis such that 1200 bytes of data (see Note above) are transported per second. For example, for a DTV video signal with a 60 Hz frame rate and no NTSC Captions, 20 bytes of DTVCC caption data are allocated in each frame.

It is very important to note that the allocation for NTSC closed-caption encoding within the DTVCC Transport Channel (sent in the Picture User Data) is included in the allocated DTVCC Caption Channel bandwidth. That is, the total DTVCC Transport Channel (consisting of the DTVCC Caption Channel and the NTSC Caption Channel) within the Picture User Data is 9600 bps. On average, a CEA-608-C datastream is allocated 960/1.001 bps, and DTVCC captions are allocated at least 8640/1.001 bps.

4.2 Pre-Allocated Bandwidth

The DTVCC Transport Channel is a fixed, pre-allocated stream which shall exist in all DTV-system bitstreams, even though captions may not be present. This ever-present bandwidth (which includes NTSC and DTVCC caption data) allows encoders to easily insert caption data into the DTV video bit-stream at the point of origin and at multiple down-stream encoding points without having to perform complex picture data processing and bandwidth re-allocation. In addition, it allows for easier decoding of the closed-caption data by the receiver since the data will always be at known locations.

The requirement to provide the DTVCC Transport Channel in a digital television stream is not part of this standard.

For transport streams with ATSC A/65C data, each program's duration is defined by an event entry in the Event Information Table as defined in ATSC A/65C.
The actual transmission speed of the bytes that are designated as DTVCC Transport Channel bytes is 19.39 Mbps. Other DTV processing systems extract these DTVCC bytes from this high rate transport stream using parsing and buffering techniques outside the scope of CEA-708-C to deliver the DTVCC content to a DTVCC processing subsystem at the fixed 9600 bps bandwidth. Accordingly, the DTVCC processing subsystems shall be able to accommodate the 9600 bps rate.

4.3 CEA-608-C Datastream

Section 4.3 describes provisions for the carriage of a CEA-608-C datastream within the DTVCC Transport Channel. The CEA-608-C datastream included in the Transport Channel provides NTSC line 21 services on NTSC video created from a received DTV signal. Section 4.3 includes requirements/recommendations for encoding and re-encoding as well as decoding. Section 9.23 provides a further discussion of issues involved in DTV to NTSC translation.

If present, the CEA-608-C datastream is located within the DTVCC Transport Layer (within the video user data of an ATSC compliant MPEG-2 video elementary stream). It exists to facilitate decoding and encoding processes required to convert the DTV video stream into an NTSC compliant video output for use with NTSC receivers and peripherals. In some situations, this allows for simpler closed-caption transcoder implementations since the DTVCC Caption Channel data stream does not have to be parsed to find the CEA-608-C datastream in the DTVCC Transport channel.

The NTSC captioning data may be transmitted in the DTVCC Transport Channel interleaved with the DTVCC Caption Channel. This data is part of the DTVCC Transport Layer but not the DTVCC Caption Channel. With the exception of the demultiplexing, NTSC Caption processing may be entirely independent of DTVCC caption decoding.

4.3.1 Transport

When an NTSC program is transcoded from NTSC to DTV for transmission in a DTV system, the entire CEA-608-C datastream (i.e. captions, text, and eXtended Data Service (XDS)) should be preserved during the encoding process (except for appropriate changes to locally generated XDS packets).

4.3.2 Generating a CEA-608-C Datastream

NOTE—For clarity, Section 4.3.2 describes both the construction and the receiver reconstruction of a CEA-608-C datastream.

Section 4.3.2 does this by describing a model for retransmitting a received CEA-608-C datastream on an NTSC output. This model describes decoder operation and requirements for CEA-708-C compliant streams; however,receiver implementation may use any process providing equivalent functionality when receiving compliant streams.

NOTE—This model turns off the run in clock; however, there is no requirement that the run in clock ever be turned off.

Two FIFO-based Field Buffers (one for **NTSC_Line21_field1_data** and one for **NTSC_Line21_field2_data**), as shown in Figure 3, hold CEA-608-C datastream byte-pairs and associated **cc_valid** flags.

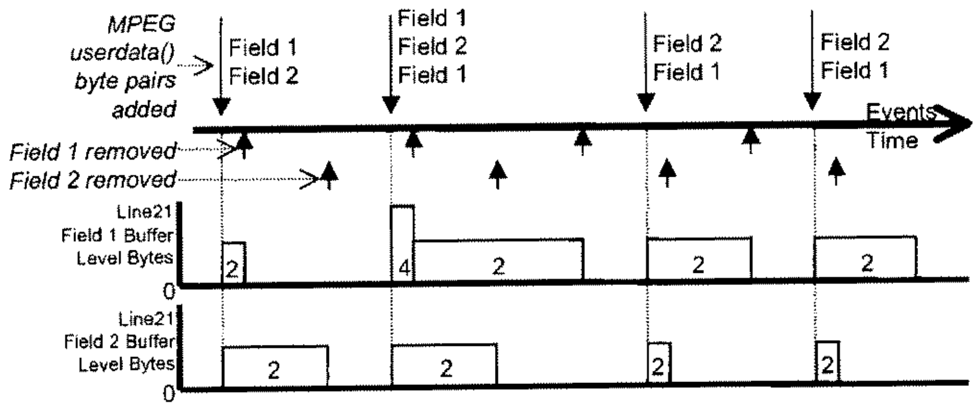

Figure 3 Example of NTSC Captioning Field Buffers

Byte pairs are received from the MPEG bitstream (at MPEG presentation time) and written into the Field Buffers in the order that they appear in the **user_data()** structure.

Bytes are removed from Field Buffer "x" at the start of NTSC field "x", and are transmitted on the NTSC signal in the order they are removed from the buffer.

If Field "x" Buffer is empty at the transmit time of NTSC field "x", a CEA-608-C datastream caption waveform should be generated for that field with **cc_data_1** = 0x80 and **cc_data_2** = 0x80 (in CEA-608-C notation, two 0x00s with odd parity).

If the byte pair at the output of the FIFO is marked "**cc_valid**==0", no CEA-608-C datastream waveform shall be generated for field "x," except that if captions are generated for field 2, CEA-608-C requires that clock run in and start bits shall also be generated for Field 1.

If an MPEG picture has no ATSC-captioning **user_data()** structure or if **process_cc_data_flag**==0, no byte pairs shall be added to the Field Buffers.

In steady-state operation, the proper Field Buffer always contains data at the start of a Field for all frame rates and repeated fields. Designers should understand that the use of frame rates of 60, 30 or 24 Hz may cause corrupted NTSC caption sequences at start-up and across splices.

CEA-608-C requires that a Field 1 waveform (run-in clock and start bit) be present whenever a Field 2 waveform is present.

When the frame rate is one of the integer values (60, 30 or 24 frames per second), the CEA-608-C datastream output will not have a direct correspondence to the input frame rate. For example, when the video is sent at exactly 30 frames per second, the length of time caption data is buffered will vary. Every 1001st MPEG frame, picture information will not be sent to the NTSC output, since only 1000 NTSC output frames are output in the period of 1001 MPEG-2 frames. When the frame rate is 30 Hz, the NTSC video encoder drops or averages pictures such that 1001 input images produce 1000 NTSC frames. If the CEA-608-C datastream rate is not reduced by sending at least one MPEG picture without a CEA-608-C datastream, the NTSC encoder cannot send all of the CEA-608-C datastream on line 21of the *generated* NTSC frames.

This data rate reduction can be done in an encoder by periodically sending **user_data()** structures without **cc_type** = 00 or 01. See Table 3. The CEA-708-C decoder does not put any data into the buffer when decoding this picture. This would normally produce a "captionless" picture (no captioning waveform on line 21). If captionless pictures occur every 1001 pictures, then, after startup, the buffer does not empty (assuming the decoder is locked to the MPEG-2 timebase).

The number of pictures with a reduced cc_type == 00 or == 01 count at the beginning of the **user_data()** structure should be minimized, but may be more frequent in portions of the program where splices have occurred. These reduced caption frequency portions of the sequence may create NTSC output with corrupted or missing caption data during steady-state operation

Encoding equipment shall not assume that decoders can buffer more than two byte-pairs (four bytes) of NTSC field 1 data and two byte-pairs (four bytes) of NTSC field 2 data, but shall send a CEA-608-C datastream at a substantially constant data rate. The size of the decoder buffer shall be at least two byte-pairs for each type, since some **user_data()** structures can have two byte-pairs of one type in them. Therefore, encoders may not assume a larger value when arranging the placement of a CEA-608-C datastream in the **user_data()** structure.

A field type's buffer receives two or four bytes at the Presentation Time of the MPEG frame, and two bytes are removed at the encoding time of the NTSC output. If there are exactly 1000 byte pairs of a type per 1001 MPEG frames, then, under steady-state operation, the buffer always has two bytes available for each NTSC output.

An NTSC frame could be encoded without a CEA-608-C datastream at start-up, if a **user_data()** structure with a reduced CEA-608-C datastream fails to fill the buffer before an NTSC frame is output. This could cause corruption in the NTSC output signal, since CEA-608-C decoding rules include special cases for repeated control codes in adjacent frames. Insertion of an empty line 21 could change these to non-adjacent control codes, changing their interpretation. This is unavoidable, and anticipated for FCC compliant NTSC caption decoders.

4.3.2.1 Alternating Transmission of CEA-608-C Datastream

When a CEA-608-C datastream is present in a stream, encoding equipment shall always include a **cc_data** structure for each field, even if a decoder is being instructed not to generate the run-in clock and start bit on one field of CEA-608-C datastream (by setting **cc_valid** to 0 for that field type).

An exception occurs when the decoded pictures produce a frame rate higher than 30/1.001, for example, when the frame rate is exactly 30 Hz. In this case, pictures with a reduced number of **cc_type** == 00 or == 01 at the beginning of the **user_data()** structure may occur to lower the average rate to no more than 30/1.001 byte pairs of each field type per second.

This allows downstream equipment to insert a legal CEA-608-C datastream without moving DTVCC caption data between video frames.

4.3.2.2 Suppressing Run-In Clock and Start Bit When No NTSC Data Present

When decoders do not have CEA-608-C **cc_data_bytes** to encode into NTSC outputs, decoders may suppress the run-in clock and start bit on the NTSC output signal, but there is no requirement that they do so. Likewise, if the entire ATSC **user_data()** structure is missing from the MPEG video sequence, decoders may suppress the run-in clock and start bit on the NTSC output signal, but there is no requirement that they do so.

4.4 DTV/MPEG-2 Picture User Data (User Bits)

The MPEG-2 video standard permits **user_data()** structures to be inserted at any of three levels (i.e., Sequence level, the Group of Pictures (GOP) level, and the Picture Data level) within the video bitstream. ATSC A/53E specifies that the DTVCC Transport Channel is only encoded at the Picture Data level (i.e., user data level 2).

Section 4.4 defines caption-related syntax elements in **user_data()** structures, and specifies semantic rules for placing caption information into the **user_data()** structure and interpreting that data.

The syntax and semantics of the captioning data structure that should be used when carrying CEA-708-C captions in transports, such as MPEG-2, are described in Table 2 (which was excerpted from ATSC A/53E Annex A Table A8.

Syntax	No. of Bits	Format
cc_data() {		
reserved	1	'1'
process_cc_data_flag	1	bslbf
zero_bit	1	'0'[1]
cc_count	5	uimsbf
reserved	8	'1111 1111'
for (i=0 ; i < cc_count ; i++) {		
one_bit	1	'1'
reserved	4	'1111'
cc_valid	1	bslbf
cc_type	2	bslbf
cc_data_1	8	bslbf
cc_data_2	8	bslbf
}		
}		

Table 2 Captioning Data Syntax (Adapted from ATSC A/53E Table A8)

4.4.1 Captioning Data Semantics

Section 4.4.1 defines the semantics of the **cc_data()** structure. See Annex B for further examples. It is an expansion of text from ATSC A/53E Annex A 5.2.3.1.

process_cc_data_flag—This flag indicates whether it is necessary to process the **cc_data**. If it is set to '1', the **cc_data** shall be parsed and its meaning processed. When it is set to '0', the remainder of the **cc_data()** structure shall still be encoded.

zero_bit—This bit shall be set to '0' to maintain backwards compatibility with previous versions of this structure.

[1] For backward compatibility, this bit shall be zero, not one.

cc_count—This 5-bit integer indicates the number of closed caption constructs following this field. It can have values 0 through 31. For common MPEG-2 frame rates, the value of **cc_count** shall be set according to the frame rate and coded picture structure (field or frame) as specified in Section 4.4.1 such that a fixed bandwidth of 9600 bps is maintained for the closed caption payload data. Sixteen (16) bits of closed caption payload data are carried in each pair of the syntax elements **cc_data_1** and **cc_data_2**.

one_bit—This bit shall be set to '1' to maintain backwards compatibility with previous versions of this structure.

cc_valid—This is a one-bit flag which indicates the validity of the two closed-caption data bytes which follow it. If **cc_valid** is set to 1, then the following two bytes of closed-caption data shall be interpreted according to Section 5 through Section 9. If **cc_valid** is set to 0, the following two closed caption bytes have no meaning; however, the bit pair **cc_type** does have meaning..

For NTSC captions, **cc_valid** shall be interpreted at the time it is inserted into the NTSC video. See Section 4.3.2.

cc_type—Denotes the type of the following two bytes of closed caption data as indicated in Table 3.

cc_type	Contents
00	NTSC line 21 field 1 closed captions
01	NTSC line 21 field 2 closed captions
10	DTVCC Channel Packet Data
11	DTVCC Channel Packet Start

Table 3 Closed-Caption Type (cc_type) Coding

If a **cc_data()** structure contains CEA-608-C datastream information (**cc_type** 00 or 01) it occurs only in the first or the first two byte-pair postions. The position following this may be a continuation of an already started DTVCC Caption Channel Packet, the start of a new DTVCC Caption Channel Packet or filler. Section 4.3.2 describes how this data may be stored in the Field Buffer model.

When **cc_type** 00 or 01 occur following **cc_type** 10 or 11 in a **cc_data()** structure, the byte pair is filler.

cc_data_1—The first of two data bytes.

cc_data_2 – The second of two data bytes. Together these form a byte pair.

Syntax elements marked "reserved" are not currently defined, have no meaning, and shall be encoded with default values. Future versions of this standard may use other values for these reserved syntax elements; therefore decoders shall ignore these reserved syntax elements, but interpret portions of the syntax that are defined.

4.4.1.1 DTVCC Caption Channel Packets in user Data

DTV Closed-Caption data is transmitted (encoded) in variable-rate, variable-sized DTVCC Caption Channel Packets (see Section 5). To allow for simple extraction and insertion of these packets without the need to fully parse their contents, the first byte-pair of each DTVCC Caption Channel Packet shall be marked using the **cc_type** = 11 syntax flag (i.e., DTVCC Caption Channel Packet Start). The remaining byte-pairs of the DTVCC packet shall be marked with the **cc_type** = 10 flag (i.e., DTVCC Caption Channel Packet Data).

A DTVCC Caption Channel Packet may continue from one **cc_data()** structure to the next. The end of the packet is indicated by either: (1) receipt of the header (**cc_valid** = 1, **cc_type** = 11) of the next DTVCC packet, or (2) receipt of a byte pair where **cc_valid** = 0 and **cc_type** = 10 or **cc_type** = 11.

These are also valid syntax for shortening Caption Channel Packets, and decoders should not assume a reset operation or other error recovery operation when they are found in the data stream.

If a syntactic element[2] is partially received when either these conditions occurs, the syntactic element shall be discarded.

> NOTE—In the DTV video compression standard, coded pictures are transmitted in a different order ("Transport" order) than they are displayed ("Display" order). Since the DTVCC captioning data extensions are part of the video coded picture constructs and follow the MPEG Video coded-picture reordering process, the order in which the captioning data extensions are transmitted is not the same order in which they will be processed by the DTVCC Decoder. The MPEG Video Decoder reorders captioning data extensions along with the pictures to which they correspond prior to the DTVCC Caption Channel Packet location and extraction method described previously. Once these captioning data are reordered, they are ready for processing by the DTVCC Decoder. See Annex D.

4.4.1.2 Padding Unused Space at the End of user_data() Structure

If DTVCC Caption Channel is or is not present in the **user_data()** structure, all encoders shall use the following structure to pad to the end of the **user_data()**: **cc_valid** = 0; **cc_type** = ##, **cc_data_1**=##, **cc_data_2**=##; where ## represents a two bit binary number. The integrity of **cc_data_x** values associated with **cc_valid** = 0 is not guaranteed.

4.4.1.3 Interpretation of cc_data() Structure

The subsections of Section 4.4.1.3 provide recommendations for encoding NTSC and DTVCC caption information.

4.4.1.3.1 Using cc_valid and cc_type within a user_data() Structure

The caption service provider has leeway in filling the **user_data()** structure. For example, as noted in Section 4.4.1 and illustrated in Section 4.4.3, the occurrence of **cc_valid** = 1, **cc_type** = 11 does not have to align with the first non-NTSC data in the **cc_data()** structure.

The value of **cc_type** is important, even if **cc_valid** = 0, since a DTVCC Caption Channel Packet can be ended if **cc_valid** ==0 and **cc_type** == 10 or 11 (see Section 4.4.1).

4.4.1.3.2 Caption Channel Packets Spanning Multiple user_data() Structures

If a single DTVCC Caption Channel Packet is to be spread across several **user_data()** structures, at least one byte pair should be included in each **user_data()** structure.

If at least one byte pair of the DTVCC Caption Channel Packet is included in the **user_data()** structure, the rest can be left unused by padding the end of the **user_data()** structure. See Annex C.

4.4.1.4 Caption Order in Progressive and Interlaced Pictures

For progressive pictures, the order of NTSC captions is only partially specified. This is because the conversion from a progressive picture to interlaced NTSC frames can be done producing a field 1 followed by a field 2, or vice versa.

4.4.2 Frame Rates

As stated in Section 4.4.1, the value of **cc_count** will vary depending on the frame rate. In any given video service, the frame rate may change (e.g. to "film-mode" during extended still picture periods). The **user_data()** construct shall be transmitted the appropriate number of times per second, and **cc_count** shall be adjusted for each such construct so that the 9600 bps DTV Transport Channel throughput is sustained no matter what the instantaneous frame rate is.

The frame rate for an MPEG-2 encoded DTV video stream is dependent upon a combination of the values for the following MPEG-2 video coding header fields: **frame_rate_code, progressive_sequence, top_field_first, repeat_first_field** and **picture_structure**; and the designation of **top field** and **bottom**

---

[2] The term "syntactic element" has specific meaning within CEA-708-C, as defined in Section 7. Other elements of the DTVCC syntax, which are not "syntactic elements," are referred to as "syntax elements."

field.These terms and fields are defined in ISO/IEC 13818-2. When defining the frame rate, these parameters also govern the **user_data()** construct transmit rate and the value of **cc_count**.
Table 4 shows the values derived for the **cc_data()** transmit rate, and allowed **cc_count** values per each frame such that the DTVCC Transport Channel data rate of 9,600 bps is maintained.

Table 4 also shows the relationship between the number of DTV data bytes vs. NTSC bytes contained in the DTVCC user data.

NTSC captioning information shall not exceed the data rate of an NTSC display (30/1.001 * 2 Bps). The value of **cc_count** of each NTSC **cc_type** shall be no greater than 30 /1.001 per second while the stream is active. This data rate may be maintained by averaging the number of each **cc_type** over 30/1.001 seconds or less. This data rate requirement shall be met across splices. For example, when two sequences with **frame_rate_value** of either 30 or 60 are spliced, the splicer could delete NTSC caption information from either the last frame of the ending sequence or the first frame of the new sequence.

If both **cc_type** 00 and **cc_type** 01 data are present in the picture **user_data()** constructs of a sequence, they shall alternate when the pictures are restored to display order.

For video sequences with **progressive_sequence**==0 for pictures with **top_field_first**==0, **cc_type** 00 shall precede **cc_type** 01.

For video sequences with **progressive_sequence**==0 for pictures with **top_field_first**==1, **cc_type** 01 shall precede **cc_type** 00.

For **picture_structure** of Top Field, **cc_type** 00 shall not be present.

For **picture_structure** of Bottom Field, **cc_type** 01 shall not be present.

For video sequences with **progressive_sequence**== 1, the ordering of **cc_type** 00 and **cc_type** 01 is arbitrary, but shall alternate. Either 4 or 6 CEA-608-C datastream **cc_data_bytes** may be included in each **user_data()** structure, subject to the data rate restriction above. DTV **cc_data_bytes** may also be present to produce a total data rate of 9600 bps, subject to the constraint on **cc_count**.

When the picture display time is extended from the nominal value of **frame_rate** indicated in the video sequence header through the repeat mechanisms in the **picture_coding_extension**, the values of **cc_count** and number of CEA-608-C datastream **cc_data_bytes** in **cc_data()** shall be modified. Values for **cc_count** and **cc_data_bytes** are shown in Table 4.

Frame_rate _value	**picture_ structure**	**progressive _frame**	**repeat_ first_field**	**top_field _first**	**cc_ count**	**NTSC cc_data _bytes**	**DTV cc_data_bytes**
60 or 59.94	11	1	0	0	10	2	18
60 or 59.94	11	1	1	0	20	4	36
60 or 59.94	11	1	1	1	30	6	54
30 or 29.97	01 or 10	0	0	0	10	2	18
30 or 29.97	11	X	0	X	20	4	36
30 or 29.97 (note)	11	1	1	X	30	6	54
24 or 23.97	11	1	0 (required)	0 (required)	25	4 or 6	46 or 44

(X = may be 0 or 1)
NOTE—For sequences with **frame_rate_value** = 30. or 29.97, **progressive_frame** ==1 and **repeat_first_field**==1 implies **progressive_sequence**==0.

Table 4 DTVCC Transport Channel Transmit Rate Parameters

To facilitate the insertion and retrieval of the CEA-608-C datastream, the **cc_data()** structure shall be formatted as follows. The CEA-608-C datastream byte pairs shall always be placed in the User Data stream before any DTVCC caption data. The CEA-608-C datastream field order (Field 1 vs. Field 2) depends on the field display order. With this scheme, once the decoder sees **cc_type** = 10 or **cc_type** = 11, indicating DTVCC caption data, it shall no longer continue searching the User Data for the CEA-608-C datastream. The CEA-608-C datastream shall follow the format and protocol as specified in CEA-608-C.

In the case of a 60.0 or 59.94 frames per second (fps) frame rate, 2 bytes of the CEA-608-C datastream are allocated per user_data extension. Except as described in Section 4.3.2, two (2) bytes of Field 1 are to be encoded every other frame, with 2 bytes of Field 2 encoded in the intervening frames. For all frame rates less than 30.0 fps, Field 1 and Field 2 captions exist within the same **cc_data()** structure.

4.4.3 Typical Video Signals

Table 5 illustrates an example of the arrangement of the closed-caption data stream in the picture **user_data()** for a typical 30 frames per second video signal frame where **cc_count** = 20. A Caption Channel Packet starts immediately after the CEA-608-C datastream byte pairs. Data between byte pairs 4 and 20 is obvious from context, the values of index 'i' are from ATSC A/53E Table 7.

i+1	Marker_bits	cc_valid	cc_type	cc_data1	cc_data2
1	1111 1	1	00	CEA-608-C datastream, field 1	CEA-608-C datastream, field 1
2	1111 1	1	01	CEA-608-C datastream, field 2	CEA-608-C datastream, field 2
3	1111 1	1	11	byte #1: DTVCC Pkt Header	byte #2: DTVCC Pkt Data
4	1111 1	1	10	byte #3: DTVCC Pkt Data	byte #4: DTVCC Pkt Data
...					
20	1111 1	1	10	byte #35: DTVCC Pkt Data	byte #36: DTVCC Pkt Data

Table 5 Aligned User Data and DTVCC Channel Packet Example

Table 6 shows the User Data frame encoding of unaligned Caption Channel Packets. The end of a previous Caption Channel Packet (that was 128 bytes in length) is in the first 4 bytes of the User Data after the CEA-608-C datastream byte pairs. The start of the first 32 bytes of a new Caption Channel Packet fills the remainder of the 40 bytes of the User Data for a 30 frames per second video signal.

i+1	cc_valid	cc_type	cc_data1	cc_data2
1	1	01	CEA-608-C datastream, field 2	CEA-608-C datastream, field 2
2	1	00	CEA-608-C datastream, field 1	CEA-608-C datastream, field 1
3	1	10	byte #125: DTV Pkt Data	byte #126: DTV Pkt Data
4	1	10	byte #127: DTV Pkt Data	byte #128: DTV Pkt Data
5	1	11	byte #1: DTVCC Pkt Header	byte #2: DTVCC Pkt Data
6	1	10	byte #3: DTVCC Pkt Data	byte #4: DTVCC Pkt Data
7	1	10	byte #5: DTVCC Pkt Data	byte #6: DTVCC Pkt Data
8	1	10	byte #7: DTVCC Pkt Data	byte #8: DTVCC Pkt Data
...				
20	1	10	byte #31: DTVCC Pkt Data	byte #32: DTVCC Pkt Data

Table 6 Unaligned User Data and DTVCC Channel Packet Example

The **marker_bits, cc_valid** and **cc_type** fields DO NOT use any part of the 9,600 bps bandwidth allocated for closed-caption data. Therefore, 9,600 bps will be maintained for the transport of closed-caption information, including the CEA-608-C datastream and DTVCC captions, within the DTV stream.

If DTVCC Caption Channel Data is present in the **user_data()** structure, it is suggested that it be inserted at the beginning of the DTVCC data area, or at the end of the data structure, but not spread across the data structure. Table 7 illustrates a non-optimal, but valid syntax.

i+1	cc_valid	cc_type	cc_data1	cc_data2
1	1	00	CEA-608-C datastream, field 1	CEA-608-C datastream, field 1
2	1	01	CEA-608-C datastream, field 2	CEA-608-C datastream, field 2
3	0	11	0xXX = END OF PACKET	0xXX = END OF PACKET
4	0	00	0x00 = padding	0x00 = padding
5	1	11	byte#1: DTVCC Pkt Header	byte#2: DTVCC Pkt Data
6	0	00	0x00 = padding	0x00 = padding
7	1	10	byte#3: DTVCC Pkt Data	byte#4: DTVCC Pkt Data
8	0	00	0x00 = padding	0x00 = padding
9	0	00	0x00 = padding	0x00 = padding
...				
20	1	10	byte#5: DTVCC Pkt Data	byte#6: DTVCC Pkt Data

Table 7 user_data_structure() Example

This is not recommended, but it is valid syntax. There may be reasons for the **user_data()** structure to have interspersed groups of **cc_valid**==0 byte pairs, such as the deletion of caption services in an edit of an MPEG stream. This example points out the possibility of unusual sequences of **cc_valid**.

4.4.4 Latency

As previously discussed in Section 4.4.1.1, the DTV picture data are transmitted in "Transport Order". This transmission path has an inherent latency associated with it, in that there can be a significant delay from the time the picture data enter a DTV encoder to the time these same data exit a DTV decoder and are available for display processing. The DTVCC data experience this same delay since they are transmitted within the picture **user_data()** bits, even though the actual DTVCC data are introduced to the encoder in "Display Order', as is the video. ATSC A/53E specifies that this latency cannot exceed 0.5 seconds.

This requirement imposes a constraint on the **vbv_delay** parameter of the MPEG-2 Picture Header bitstream construct. This value is the Video Buffering Verifier (VBV) Delay. The VBV is a hypothetical decoder that is conceptually connected to the output of the DTV encoder. Its purpose is to provide a constraint on the variability of the data that an encoder or editing process may produce.

vbv_delay is a 16-bit unsigned integer in the Picture Header. For constant bit rate operation, **vbv_delay** is used to set the initial occupancy of the decoder's video buffer at the start of play so that the decoder's buffer does not overflow or underflow. **vbv_delay** measures the time needed to fill the VBV buffer from an initially empty state at the target bit rate, R, to the correct level immediately before the current picture is removed from the buffer.

The value of **vbv_delay** is the number of periods of the 90 kHz system clock that the VBV shall wait after receiving the final byte of the picture start code. For a 0.5 second delay, the **vbv_delay** value is 45,000 (i.e., 45,000 cycles / 90,000 cycles per second = 0.5 seconds). If **vbv_delay** <= 45,000 for a picture start code, it means that the closed captions in that picture data will experience a delay of less than 0.5 second in the decoder buffer.

4.4.4.1 Zero Delay Progressive Sequences

For non-interlaced (progressive) pictures, decoders may produce an NTSC output with either field order. For example, if an MPEG picture with **frame_rate_value** of 30 in a progressive sequence includes valid

cc_type = 00 followed by **cc_type** = 01 in **user_data()**, then the decoder can produce an NTSC output with **cc_type** = 00 on the first (NTSC field 1) field derived from the progressive picture, and the **cc_type** = 01 on the second (NTSC field 2) field, or may put previous caption data on the first (NTSC field 2) field derived from the progressive picture and the **cc_type** = 00 data on the second (NTSC field 1) field. Either of these arrangements may be considered as "zero delay".

> NOTE—Reversal of the output order of **cc_type** 00 and 01 relative to the Input order is not recommended, but is not explicitly forbidden. That is, the latency for **cc_type** 00 may differ from the latency of **cc_type** 01.

4.4.4.2 Zero Delay Interlaced Sequences
For interlaced pictures, the **cc_type** 01 is associated with a Top Field, and **cc_type** = 00 with a Bottom Field. DTV decoders may translate Top Fields to NTSC field 1 or to NTSC field 2. If Top Fields are translated to field 1, then "zero delay" shall be defined as inserting the captioning data into the field immediately subsequent to the field in which it is transmitted. If Top Fields are translated to field 2, then "zero delay" shall be defined as inserting the captioning data into the field in which it was transmitted.

4.4.4.3 End-to-end Delay
The total delay for caption information relative to the video should be maintained at or below four NTSC frame times (that is, 4/30 second) for 60 Hz based signals and at or below four 25 Hz frames when associated with 50 Hz based frame rates. These delays include those of caption encoders, remultiplexers or other processing equipment, and decoders.

4.4.4.4 Encoding Delay
The delay for caption information relative to the video for encoders should be maintained at or below two NTSC frame times (that is, 2/30 second) for 60 Hz based signals and at or below two 25 Hz frames when associated with 50 Hz based frame rates.

4.4.4.5 Decoding Delay
The total delay for caption information relative to the video for decoders should be maintained at or below 2/30 second for 60 Hz based signals and at or below two 25 Hz frames when associated with 50 Hz based frame rates.

4.4.4.6 Intermediate Processing Delay
Intermediate processors, including those that combine caption streams, remultiplex video signals and otherwise process video independent of caption information should not introduce any systematic advance or delay of captions, unless the adjustment is introduced to compensate for a constant delay component elsewhere in the processing system.

4.5 Caption Service Metadata
Each caption service may have metadata items that describe the service. These metadata items may be carried "out of band" to the caption services themselves, such as in an MPEG-2 transport descriptor. The requirements for metadata transmission are not within the scope of CEA-708-C; for ATSC transmission they are defined in ATSC A/65C.

The service metadata items are defined as follows:

> NUMBER OF SERVICES: the total number of caption services (1-16) present over some transport-specified time period. There shall be a maximum of 16 separate services present.
>
> > NOTE—The signaling of this metadata need not apply to each **cc_data()** structure, but might be more useful in some applications as an aggregate of the caption services over some well-understood time period for the application, such as a program event duration.

For each caption service, the following metadata can be conveyed:

TYPE OF SERVICE: {608 | 708} the service contains a CEA-608-C datastream or CEA-708-C bytes. There shall be at most one CEA-608-C datastream signaled. The CEA-608-C datastream itself signals

the individual CEA-608-C caption channels. This is in contrast to 708 services, which map 1:1 to the 708 caption service numbers and additional metadata items.
When the TYPE OF SERVICE is 708, then the following 708-related metadata should be conveyed:

> SERVICE NUMBER: the service number as found in the 708 caption service block header (1-31). This field provides the linkage of the remaining metadata to a specific 708 caption service.
>
> LANGUAGE: the dominant language of the caption service. It is recommended that this be encoded from ISO 639-2:1998.
>
> > NOTE—An encoding is recommended to ensure an appropriate breadth of languages are unambiguously supportable. When a caption service contains multiple languages, this metadata item should indicate the predominant language most viewers would select.
>
> DISPLAY ASPECT RATIO: {4:3 16:9} the display aspect ratio assumed by the caption authoring in formatting the caption windows and contents.
>
> EASY READER: this metadata item, when present, indicates that the service contains text tailored to the needs of beginning readers.
>
> CAPTION CHANNEL SERVICE DIRECTORY: This is the aggregate of all metadata described above, for all services present over some period of time.

The ATSC Caption Channel Service Directory is contained in the **caption_service_descriptor()**.The **caption_service_descriptor()** is defined by ATSC A/65C, and can be carried in the PMT and the EIT of the MPEG-2 transport stream.

4.5.1 Decoder Processing of Metadata

If metadata is available through multiple structures, these structures shall be consistent or have well defined priority rules. For example, in ATSC streams the Caption Service Descriptors are encoded in the PMT and, when present, the EIT. These descriptors shall match and change in a consistent way. The decoder only needs to acquire this information from one location. FCC requlations require decoders to acquire service descriptors from the EIT only. Acquiring all program information from a single, concise table facilitates fast tuning.

Decoders shall be capable of selecting any of the (up to sixteen) CEA-708-C caption services described in the **caption_service_descriptor** as currently defined. Selection of CEA-608-C caption channels is optional.

Decoders that allow selection of NTSC caption services should allow any one of the four caption channels (CC1-CC4) to be selected, even if no **caption_service_descriptor()** is present. This allows decoders to continue access to CC2 and CC4.

The presence or absence of CEA-608-C metadata entries shall have no effect on the encoding of the CEA-608-C datastream in the **cc_data()** structure onto NTSC output signals.

In the absence of metadata describing a caption service, the decoder should decode CEA-708-C captions as if metadata was present with Display Aspect Ratio value of 16:9. In the event a decoder cannot render a caption window on the display device within the device's safe title area, the caption window can be completely disregarded. It is preferred that decoders transforming the aspect ratio of the received video (letterboxing a 16:9 image for 4:3 display, for example) apply similar transforms to received caption material to achieve captions coordinated with the displayed image format.

The grid locations for overlaying caption windows on an image are assumed to be from the content provider's perspective and refer to the transmitted frame. It is preferred that any image format alterations

performed in the decoder (for example, 'letterboxing') should be similarly applied to the DTVCC window positioning grid for the accompanying caption material.

4.6 User Data Syntax

ATSC A/53E Annex A Table 8 defines the picture **user_data()** syntax, and the use of that syntax in transmission of DTVCC streams. CEA-708-B Section 4.4 addresses DTV/MPEG-2 Picture User Data (User Bits).

Section 4.6 describes how implementation choices can allow for future expansion, and how to interpret various ATSC specified (but not defined) syntax elements.

The **ATSC_identifier** is used to determine which **user_data()** structure carries the ATSC information. MPEG allows multiple **user_data()** structures per picture, each beginning with a **user_data_start_code**. If non-ATSC **user_data()** is present in a stream, it will not begin with **ATSC_identifier**.

The **user_data_type_code** is set to 0x03 by ATSC. At some future date, ATSC may define other values for **user_data_type_code**.

The values for **marker_bits** are defined in ATSC A/53E Annex A Table 8.

The **additional_user_data** can have any value, though **start_code** emulation is forbidden based on section 6.2.1 of ISO/IEC 13818-2, and the definitions in ATSC A/53E Annex A Table 8.

4.6.1 Picture user_data() Syntax

4.6.1.1 Undefined Syntax Elements in user_data() Structure

Any syntax elements in ATSC A/53E Annex A Table 7, not currently defined, have no meaning, and should be encoded with default values. Future encoders may produce signals based on extensions of the specifications, therefore decoders should ignore syntax elements not defined by current standards, but interpret portions of the syntax that are defined.

4.6.1.2 ATSC_identifier and user_data_type_code

CEA-708-C decoders should ignore **user_data()** constructs which do not have both the correct **ATSC_identifier** and **user_data_type_code** == 0x03. Encoded streams shall not have more than one construct with the correct **ATSC_identifier** and **user_data_type_code** == 0x03 per **picture_header**.

4.7 Shortened Caption Channel Packets

There are two methods which end a Caption Channel Packet. Neither of these methods uses the **packet_size** value determined from the first byte of the Caption Channel Packet.

If a syntactic element is partially received when either of these conditions occurs, it shall be discarded. This situation might arise when a long packet is being sent at the time of a splice; there is no way to continue the long packet into the new stream, so one of the two shortening techniques may be used.

4.8 Independent Control of Audio and Caption Services

The caption data shall be an independent service from the audio service. It is intended that hearing and deaf viewers may share the same living room, watching the same program. Each viewer should be able to experience the service that they use (audio or captions) independent of adjustment of the other; for example, the choice of audio language or caption language should be independently selectable. Devices using captioning data shall provide a mode in which no modification of audio settings (including language) occurs when captioning is enabled, changed or disabled. In addition, no modification of the captioning data display shall occur when audio settings (including language) are changed. However, other modes, such as 'caption when mute enabled' may be implemented.

5 DTVCC Packet Layer

The DTVCC Caption Channel data are framed as packets within the DTVCC Transport Channel prior to encoding. This packet structure is completely defined within CEA-708-C. (Error detection, error correction, compression, and other low-level transport overhead issues are handled in the DTV transmission layers, and thus fall within the purview of the ATSC and MPEG-2 standards).

The DTVCC Packet Layer is defined by a variable-size Caption Channel Packet of **n** bytes of closed captioning data where **n** <= 128, and is an even number. Caption Channel Packets are coded in the **user_data()** extensions of the coded pictures in the DTV stream. The beginning and ending of each Caption Channel Packet in the User Data is indicated by the syntax defined in Section 4.1.

Each Caption Channel Packet consists of a one byte header and **n-1** bytes of data, where **n** is the total packet size. (See Figure 4.) The header contains the **packet_size** code and **sequence_number**.

b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0	
Seq. No.		Packet Size (n/2)						Packet Header
Caption Channel Data Byte 1								Packet Data
Caption Channel Data Byte 2								
. . .								
Caption Channel Data Byte n-1								

Figure 4 Caption Channel Packet

The syntax for the **caption_channel_packet** is shown in Figure 5 .

	No. Of bits	Mnemonic
caption_channel_packet() {		
sequence_number	2	uimsbf
packet_size_code	6	uimsbf
For (i = 0; i < packet_data_size; i++) {		
Packet_data[i]	8	bslbf
}		
}		

Figure 5 DTVCC Caption Channel Packet Syntax in ATSC A/53E Notation

The variables, **sequence_number, packet_size_code** and **packet_data** are defined as follows:

sequence_number is a two-bit (b6 - b7) rolling sequence (from 0 to 3) which may be used by receivers to detect the loss of packets. When packet loss is detected, any partially accumulated data from the previously received packet shall be discarded, and Reset command processing shall be performed for each existing Caption Service. See Section 8.9.5. Decoding shall resume with the first Service Block encountered in the new Channel Packet. See Section 6.2.

Detection of data loss may be accomplished by using discontinuities in the **sequence_number** to detect loss of synchronization. Because there are only four possible values for **sequence_number**, not all data disruptions cause discontinuities in the sequence. For example, if the number of lost packets is a multiple of four, the count appears to be continuous, and packet loss is not detected.

Decoder designers may wish to implement an enhanced discontinuity detection method. At the start of each Caption Channel Packet, the command parser state is checked to assure that all syntactic elements from the previous packet have been completed, and no partially accumulated data remains. If data remains in the parser, then a discontinuity may be assumed to have occurred, since DTVCC Caption Channel Packets ought to only contain complete syntactic elements. See Section 6.3. The decoder would then perform Reset command processing for each Caption Service.

packet_size_code—Contained in the lower 6 bits of the header (b0 – b5), these bits define the **packet_size** variable using the algorithm:

If **packet_size_code** == 0, then **packet_size** is 128;

Otherwise, **packet_size** is determined by the formula:

packet_size = 2 * **packet_size_code**

packet_data_size = ***packet_size*** -1

packet_data—This array of bytes is organized into Service Blocks, as described in Section 6.2.4. The processing of over-length packets is undefined. Encoder and processing equipment manufacturers should be aware that if the amount of Caption Channel Packet data is greater than indicated by **packet_size**, decoders may ignore the extra data.

6 DTVCC Service Layer

The DTVCC Caption Channel is divided into a set of logical sub-channels or "Services." The Service Layer defines the headers for the caption data channel Service Numbers, Service Types, and Service Attributes.

Receivers use the information in this layer to route the Service Data to the appropriate internal processing modules.

6.1 Services

The Caption Channel Services are of two types: 6 Standard services (Service #1 through Service #6) and 57 Extended services (Service #7 through Service #63) allowing for 63 total services. Service #0 shall not be used.

Service #1 is designated as the Primary Caption Service. This service contains the verbatim or near-verbatim, captions for the primary language being spoken in the accompanying program audio.

Service #2 is designated as the Secondary Language Service. This service contains captions in a secondary language, which are translations of the captions in the Primary Caption Service or may be captions of a Secondary audio service.

The other service sub-channels are not pre-assigned. It is up to the discretion of the individual caption provider to utilize the remaining service channels.

6.2 Service Blocks

Caption Channel Service Blocks provide the structure for the asynchronous time division multiplexing of Service data within the DTV Closed-Caption Channel. Service data are time-division-multiplexed into the appropriate time slots in the Caption Channel Packets as required. See Section 6.2.5.

Caption providers and encoding equipment algorithms govern the frequency, priority, and bandwidth consumption for each individual Service carried in these Service Blocks.

Each Service Block consists of a 1 or 2 byte header followed by 1 to 31 bytes of data. See Figure 6.

Service Block Header (1 or 2 bytes)
Service Block Data Byte 1
. . .
Service Block Data Byte m (m <= 31)

Figure 6 Service Block

The Service Block syntax is shown in Figure 7.

	No. Of bits	Mnemonic
Service_block() {		
service_number	3	uimsbf
block_size	5	uimsbf
if (service_number == b'111' && block_size != 0){		
null_fill	2	'00'
extended_service_number	6	uimsbf
}		
if (service_number != 0) {		
for (i = 0; i < block_size; i++) {		
Block_data	8	bslbf
}		
}		
}		

Figure 7 Service Block Syntax

6.2.1 Standard Service Block Header

A one-byte Standard Service Block Header consists of 2 parts: the Service Number **(sn)**, and the Service Block Size **(bs)**. See Figure 8. The Service Number is defined in the 3 high-order bits **(sn_0 - sn_2)**, and the Service Block Size is defined in the 5 low-order bits **(bs_0 - bs_4)**.

The service Block Size value ranges from 1 to 31 and indicates the number of bytes following the header. A Standard Service Block Size of 0 shall not be used, except in a Null Service block Header. See Section 6.2.3.

b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0
sn_2	sn_1	sn_0	bs_4	bs_3	bs_2	bs_1	bs_0

Figure 8 Service Block Header

6.2.2 Extended Service Block Header

If more than 6 simultaneous services are required within the Caption Channel, a 2-byte Extended Service Block Header is used, as shown in Figure 9. The first byte of this Extended Header has the same format as the Standard Service Block Header, with the exception that the 3 high order bits **(sn2 – sn0)** are each fixed to a value of 1. This value of "7" in the Standard Service Number bits signals the use of an Extended Service Block Header. The 2nd header byte contains an Extended Service Number **(sn_0 - sn_5)** whose value may range from 7 to 63. Extended service numbers less than 7 are not permitted, since they can already be specified by the Standard Service Block Header.

The Block Size value in the lower 5 bits **(bs_0 - bs_4)** of the first byte of the Extended Service Block Header signals the number of Service Block Data bytes that follow the Extended Service Block Header.

b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0
1	**1**	**1**	bs_4	bs_3	bs_2	bs_1	bs_0
0	**0**	sn_5	sn_4	sn_3	sn_2	sn_1	sn_0

Figure 9 Extended Service Block Header

6.2.3 Null Service Block Header

The NULL Service Block Header is shown in Figure 10. All bits are set to 0. The appearance of a NULL Service Block Header indicates that there are no more Service Blocks in the Caption Channel Packet for

the decoding hardware to process. This is equivalent to a Service Block with **service_number** == 0 and **block_size** == 0. When **service_number** == 0, **block_size** shall be 0.

b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0
0	0	0	0	0	0	0	0

Figure 10 Null Service Block Header

6.2.4 Service Block Data

Within each Service Block, the header may be followed by up to 31 bytes of Service Data. This data consists of Coded Caption Data and Data Interpretation. See Section 8.

Service Data are demultiplexed from the Service Blocks and routed to the appropriate service processors in the DTV receiver's decoder. This separation creates the individual byte streams for each of the services, which are handed off to the DTVCC Coding Layer.

6.2.5 Service Blocks within Caption Channel Packets

Service Blocks are time division multiplexed and inserted sequentially into the Caption Channel Packets.

Service Blocks shall not cross Caption Channel Packet boundaries. If a service requires more data than the current Channel Packet allows, the Service Block Data shall be truncated to fit the current packet. Any remaining data bytes shall be embedded in a new Service Block (with Header), and placed within a subsequent Caption Channel Packet.

Figure 11 shows an example of a Caption Channel Packet containing two Standard Service Blocks and one Extended Service Block. The Caption Channel Packet size is 20 bytes and the Packet Sequence Number is 2.

The double lines indicate the start of byte pairs. The heavy double line indicates that the byte pair is marked "DTVCC Channel Packet Start".

A Null Service Block Header (Figure 10) shall be inserted as the last Service Block in the Channel Packet if space permits.

Packet Encoding equipment should null fill (i.e., zero out) the Channel Packet buffer before any Service Blocks, which insures that a Null header is always present in Channel Packets that are not completely filled.

Syntax:	msb							lsb	Packet Byte:
Pkt Size: 20/2, Seq# 2	1	0	0	0	1	0	1	0	0 (Packet Header)
Svc #: 1, BlkSz: 3	0	0	1	0	0	0	1	1	1 (Service 1)
									2
									3
									4
Svc #: 6, BlkSz: 4	1	1	0	0	0	1	0	0	5 (Service 6)
									6
									7
									8
									9
Ext Svc, BlkSz: 8	1	1	1	0	1	0	0	0	10 (Ext. Service 21)
Svc#: 21	0	0	0	1	0	1	0	1	11
									12
									13
									14
									15
									16
									17
									18
									19

Figure 11 Service Blocks in a Caption Channel Packets (Example)

6.3 Transport Constraints on Encapsulating Caption Data

The Caption Channel Packet and Service Block layers provide two mechanisms for encapsulating caption data. Service Block boundaries are asynchronous with respect to the syntactic elements defined in Section 7. One service may be specified for several service blocks within a Caption Channel Packet, and, therefore, a syntactic element started in one service block may be finished in a subsequent block.

The Service Block and the Caption Channel Packet are both asynchronous with relation to the **user_data()** structure.

Unlike the Service Block, the DTVCC Caption Channel Packet, described in Section 5, is a framing syntax, containing Caption Channel data. The start of a DTVCC Caption Channel Packet is a point where decoding of Caption Channel data may be resumed. For proper parsing, the Caption Channel data shall start with the first byte of syntactic element code.

Caption service providers shall not cross DTVCC Caption Channel Packet boundaries with syntactic elements. Every syntactic element shall be entirely contained in a single DTVCC Caption Channel Packet. DTVCC Caption Channel Packets shall begin with a service block header, and, for each service present, the first byte of **Block_data** shall begin a syntactic element.

Aligning syntactic elements to Service Blocks as well as DTVCC Caption Channel Packets may simplify stream creation, and caption service encoders may do this by design; however, decoder designers should not assume that all streams have this additional alignment.

Caption service providers should be aware that decoders may reset if syntactic elements are left incomplete at the end of a DTVCC Caption Channel Packet, but that there is no requirement for decoders to reset. If incomplete syntactic elements remain, unexpected information may be displayed, and the decoder may be left in an indeterminate state.

Caption service providers shall not send Service Blocks with **service_number** == 0 with a **block_size** other than 0. However, if Service Blocks are received with **service_number** == 0 and **block_size** != 0, this should be interpreted as a null service block header regardless of the value of **block_size**. The

parser should discard any remaining bytes in the current packet, and resume decoding the next Service Block Header at the beginning of the next Channel Packet.

7 DTVCC Coding Layer - Caption Data Services (Services 1 - 63)

The DTVCC Coding Layer defines the numerical codes that are used for caption characters and symbols, caption commands, and Code-Space control.

Caption data and control information are transmitted as a stream of byte-codes carried in a sequence of Service Blocks for a particular service. Most byte-codes specify window commands or alphanumeric characters. Some byte-codes are extension codes that cause the bytes that follow to be interpreted in a special way. These code sequences are referred to as syntactic elements. Each syntactic element begins with a numerical code.

The following sections describe the construction of these fixed and variable length syntactic elements, which can range in length from 1 to 31 bytes.

A Caption Service decoder parses the stream of bytes into a sequence of syntactic elements. Defined syntactic elements are interpreted and used as specified in the DTV Interpretation Layer. Unused syntactic elements are parsed and interpreted as defined in the subsections of Section 7.

7.1 Code Space Organization

Consistent with ANSI INCITS 4 and ISO 2022, the 256 position code space is divided into four code groups: the CL, GL, CR and GR. See Table 8 and Table 9. Each group contains a standard code set and an extended code set.

a) The CL group contains the 32 addressable codes from 0x00 to 0x1F. The C0 (a subset of the ASCII, ANSI INCITS 4 Miscellaneous Control Codes) and C2 (Extended Miscellaneous Control Codes) code sets are mapped to this space.
b) The GL group contains the 96 addressable codes from 0x20 to 0x7F. The G0 (a slightly modified version of the ANSI INCITS 4 ASCII Printable Character set) and G2 (Extended Control Code Set 1) code sets are mapped to this space.
c) The CR group contains the 32 addressable codes from 0x80 to 0x9F. The C1 (Caption Control Codes) and C3 (Extended Control Code Set 2) code sets are mapped to this space.
d) The GR group contains the 96 addressable codes from 0xA0 to 0xFF. The G1 (ISO 8859-1 Latin 1 Characters) and G3 (future character and icon expansion) code sets are mapped to this space.

<table>
<tr><td>b7-b4:</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>A</td><td>B</td><td>C</td><td>D</td><td>E</td><td>F</td></tr>
<tr><td>b3-b0 0</td><td colspan="2" rowspan="16">CL
(C0, C2)
Set</td><td colspan="6" rowspan="16">GL
(G0, G2)
Set</td><td colspan="2" rowspan="16">CR
(C1, C3)
Set</td><td colspan="6" rowspan="16">GR
(G1, G3)
Set</td></tr>
<tr><td>1</td></tr>
<tr><td>2</td></tr>
<tr><td>3</td></tr>
<tr><td>4</td></tr>
<tr><td>5</td></tr>
<tr><td>6</td></tr>
<tr><td>7</td></tr>
<tr><td>8</td></tr>
<tr><td>9</td></tr>
<tr><td>A</td></tr>
<tr><td>B</td></tr>
<tr><td>C</td></tr>
<tr><td>D</td></tr>
<tr><td>E</td></tr>
<tr><td>F</td></tr>
</table>

Table 8 DTVCC Code Space Organization

	C0		G0						C1		G1					
b3-b0 \ b7-b4	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	EXT1	SP	0	@	P	`	p	CW0	SPA	NBS	°	À	Ð	à	ð
1			!	1	A	Q	a	q	CW1	SPC	¡	±	Á	Ñ	á	ñ
2			"	2	B	R	b	r	CW2	SPL	¢	²	Â	Ò	â	ò
3	ETX		#	3	C	S	c	s	CW3		£	³	Ã	Ó	ã	ó
4			$	4	D	T	d	t	CW4		¤	´	Ä	Ô	ä	ô
5			%	5	E	U	e	u	CW5		¥	µ	Å	Õ	å	õ
6			&	6	F	V	f	v	CW6		¦	¶	Æ	Ö	æ	ö
7			'	7	G	W	g	w	CW7	SWA	§	·	Ç	×	ç	÷
8	BS	P16	(	8	H	X	h	x	CLW	DF0	¨	¸	È	Ø	è	ø
9			)	9	I	Y	i	y	DSW	DF1	©	¹	É	Ù	é	ù
A			*	:	J	Z	j	z	HDW	DF2	ª	º	Ê	Ú	ê	ú
B			+	;	K	[	k	{	TGW	DF3	«	»	Ë	Û	ë	û
C	FF		,	<	L	\	l	|	DLW	DF4	¬	¼	Ì	Ü	ì	ü
D	CR		-	=	M	]	m	}	DLY	DF5	-	½	Í	Ý	í	ý
E	HCR		.	>	N	^	n	~	DLC	DF6	®	¾	Î	Þ	î	þ
F			/	?	O	_	o	♪	RST	DF7	¯	¿	Ï	ß	ï	ÿ

Standard Code

b3-b0 \ b7-b4	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0			TSP	█							CC					
1			NBTSP	'												
2				'												
3				"												
4				"												
5			…	•												
6								⅛								
7								⅜								
8								⅝								
9				™				⅞								
A			Š	š				│								
B								┐								
C			Œ	œ				└								
D				℠				─								
E								┘								
F				Ÿ				┌								
	C2		G2						C3		G3					

Extended Code

Table 9 DTVCC Code Set Mapping

NOTE—This section lists all required and optional caption codes. Refer to Section 9.4 for a list of required codes for a minimum DTVCC receiver implementation.

7.1.1 Extending the Code Space

The characters in the extended code sets of the CL, CR, GL, and GR code groups are accessed using the **EXT1** code (0x10) in the C0 code set (Miscellaneous Control Codes). Normally (i.e., without extending the code space) the base codes in the four code groups (CL, GL, CR, and GR) represent the characters, control codes, and commands in the C0, C1, G0, and G1 code sets. By prefixing the codes in the code

groups with the **EXT1** code (and thus forming the start of a two-byte code), the symbols in the extended C2, C3, G2, and G3 code sets are referenced; that is, each character in these code sets require the transmission of two codes (i.e., **EXT1** plus a base code) in order to be referenced. **EXT1** shall start every 2-byte extended code sequence for the C0, C1, G0 and G1 code sets. See 7.1.4 for other codes for indicating two byte codes.

EXT1 is only active for the two-byte extended code sequence in which it exists.

For example, to generate the closed-caption symbol (CC) in the G3 code set, the following sequence shall be specified: 0x10, 0xA0 (**EXT1**, (CC)).

The code space may be extended further (for future use) to handle other character sets not included in the language sets listed in Section 7.1.7. This extension capability anticipates code sets which require 16-bit character code addressing, such as characters to implement Chinese and Japanese. The **P16** code in the C0 Miscellaneous Control Code set is used for this purpose. When a decoder encounters the **P16** code, it uses the succeeding two bytes to address characters in a 16-bit code set.

7.1.2 Unused Codes

Syntactic elements in which the first code is undefined, reserved, or in which the first byte is **EXT1** and the second code is undefined or reserved are reserved for future use. Decoders encountering any of these syntactic elements should adhere to the size (e.g., 1 byte, 2 byte, 3 byte and variable length) characteristics of these syntactic elements, as indicated in the following sections. This assures that future extensions to the coding scheme can be ignored by decoders which do not support them.

7.1.3 Numerical Organization of Codes

The following subsections detail the mapping of the individual code sets (C0, C1, C2, C3, G0, G1, G2, and G3) of the DTVCC code space.

7.1.4 Code Set C0 - Miscellaneous Control Codes

The C0 code set contains the 32 addressable codes from 0x00 to 0x1F.

a) Codes 0x00 through 0x0F are single byte codes.
b) Codes 0x10 through 0x17 are two-byte codes.
c) Codes 0x18 through 0x1F are three-byte codes.

The **NUL**, **BS**, **FF**, and **CR** codes are preserved from the standard ASCII control code set. The **ETX** code is also from the ASCII control code set, but it has a special use in that it is required at the end of a caption text segment to terminate the segment when the segment is not immediately followed by another caption command (see Section 8.10.5). The **EXT1** code is used to extend the DTVCC code space (see Section 7.1.1). The **P16** code is used as a further code space extension for 16-bit character sets.

	C	0
b3-b0 \ b7-b4	0	1
0	NUL	EXT1
1		
2		
3	ETX	
4		
5		
6		
7		
8	BS	P16
9		
A		
B		
C	FF	
D	CR	
E	HCR	
F		

Table 10 C0 Code Set

The undefined codes in Code Set C0 shall be parsed as follows, but shall not affect the caption display. Unused codes within the range of 0x00 to 0x0F shall be skipped. All unused codes from 0x11 to 0x17 shall be skipped along with the following byte (2 byte sequence). All unused codes from 0x18 to 0x1F shall be skipped along with the following two bytes (3 byte sequence).

The C0 Control Codes "CR", "HCR" and "FF" in Table 8 should perform the following operations:

Carriage Return (CR) moves the current entry point to the beginning of the next line. If the next line is below the visible window, the window "rolls up" as defined in CEA-608-C Section 7.4. If the next line is within the visible window and contains text, the cursor is moved to the beginning of the line, but the pre-existing text is not erased. See Sections 8.4.9.2, 8.4.9.3 and Table 21 for further definition of this code.

Horizontal Carriage Return (HCR) moves the current entry point to the beginning of the current line without line increment or decrement. It shall erase all text on the line. It is similar to moving the pen location to the "nth" location on the line and issuing "n" BS (BackSpace) commands, (where "n" is the number of filled-character positions on the current line). The BS command cannot erase the last character in a line, but HCR should erase the last character, similar to the DER command in CEA-608-C.

Refer to the "Carriage Return Behavior" column in Table 21 to determine the location of the "beginning of the current line". ("Line" means "Row" for Left->Right or Right->Left print directions, and "Column" for Top->Bottom and Bottom->Top print directions).

Form Feed (FF) erases all text in the window and moves the cursor to the first character position in the window (0,0). This is equivalent to specifying the window in a ClearWindows (CLW) command, followed by SetPenLocation (SPL) (0, 0).

Codes 0x00 through 0x0F at the start of a syntactic element indicate that the byte forms a single byte syntactic element.

Codes 0x10 through 0x17 at the start of a syntactic element indicate that the byte and the following byte form a two-byte syntactic element.

The code 0x10 (EXT1) is the first byte of a two-byte syntactic element, as specified in Section 7.1.8 through Section 7.1.11.

7.1.5 C1 Code Set - Captioning Command Control Codes

The C1 code set contains the 32 addressable codes from 0x80 to 0x9F. See Table 11. This set contains Window Creation commands, Character Attributes, etc. The use of these codes and the DTVCC Captioning Commands are detailed in Section 8.10.

	C	1
b3-b0 \ b7-b4	8	9
0	CW0	SPA
1	CW1	SPC
2	CW2	SPL
3	CW3	
4	CW4	
5	CW5	
6	CW6	
7	CW7	SWA
8	CLW	DF0
9	DSW	DF1
A	HDW	DF2
B	TGW	DF3
C	DLW	DF4
D	DLY	DF5
E	DLC	DF6
F	RST	DF7

Table 11 C1 Code Set

A Window command code is a single byte that may be followed by several parameter bytes.

A code from the C1 Code Set at the start of a syntactic element indicates that the byte, in some cases combined with following bytes in the stream, form a syntactic element with length as defined in this section and in the subsections of Section 8.10.5.

7.1.5.1 Code Set C1 Unused Commands

Undefined Window command codes 0x93 to 0x96 are one byte command and are reserved.

7.1.6 G0 Code Set - ASCII Printable Characters

The G0 code set contains the 96 addressable codes from 0x20 to 0x7F. See Table 12.This set provides the ANSI INCITS 4 ASCII printable characters with the substitution of the music note character for the ASCII DEL character.

The underline character (0x5F) in the G0 Code set should be used as a substitute for unsupported graphic symbols in the G3 Code set. See Section 7.1.9.

	G0					
b3-b0 \ b7-b4	2	3	4	5	6	7
0	SP	0	@	P	`	p
1	!	1	A	Q	a	q
2	"	2	B	R	b	r
3	#	3	C	S	c	s
4	$	4	D	T	d	t
5	%	5	E	U	e	u
6	&	6	F	V	f	v
7	'	7	G	W	g	w
8	(	8	H	X	h	x
9	)	9	I	Y	i	y
A	*	:	J	Z	j	z
B	+	;	K	[	k	{
C	,	<	L	\	l	|
D	-	=	M	]	m	}
E	.	>	N	^	n	~
F	/	?	O	_	o	♪

Table 12 G0 Code Set

A code from the G0 Code Set at the start of a syntactic element indicates that the byte forms a single-byte syntactic element.

7.1.7 G1 Code Set - ISO 8859-1 LATIN-1 Character Set

The G1 code space contains the 96 addressable codes from 0xA0 to 0xFF. See Table 13. This set consists of the ISO 8859-1 Latin-1 character set, also known as the Windows/ANSI character set. This set, when used in conjunction with the ASCII set, provides all the characters needed to encode text in Danish, Dutch, Faeroese, Finnish, French, German, Icelandic, Irish, Italian, Norwegian, Portuguese, Spanish and Swedish. Many other languages can be written with this set of letters, including Hawaiian, Indonesian/Malay, and Swahili. ISO 8859-1 also extends the ASCII set with additional miscellaneous punctuation and mathematical signs.

Code **NBS** (0xA0) represents a non-breaking space. When *wordwrap* = 0, this character is the same as a space character. Interpretation of this code when *wordwrap* = 1, is reserved.

b3-b0 \ b7-b4	A	B	C	D	E	F
0	NBS	°	À	Ð	à	ð
1	¡	±	Á	Ñ	á	ñ
2	¢	²	Â	Ò	â	ò
3	£	³	Ã	Ó	ã	ó
4	¤	´	Ä	Ô	ä	ô
5	¥	µ	Å	Õ	å	õ
6	¦	¶	Æ	Ö	æ	ö
7	§	·	Ç	×	ç	÷
8	¨	¸	È	Ø	è	ø
9	©	¹	É	Ù	é	ù
A	ª	º	Ê	Ú	ê	ú
B	«	»	Ë	Û	ë	û
C	¬	¼	Ì	Ü	ì	ü
D	-	½	Í	Ý	í	ý
E	®	¾	Î	Þ	î	þ
F	¯	¿	Ï	ß	ï	ÿ

Table 13 G1 Code Set

7.1.8 G2 Code Set - Extended Miscellaneous Characters

The G2 code set contains the extended miscellaneous characters. See Table 14. Characters in the G2 space are transmitted by preceding them with the control character **EXT1** (0x10) from the C0 character set. Following the **EXT1** extender prefix, the G2 characters are addressed with the base in the range 0x20 – 0x7F.

The unshaded characters in the G2 table below are those which exist in the Microsoft Windows character map. They are positioned in this code set to reflect their relative positioning in the MS Windows map. In Windows, these characters are mapped to the code set in the range 0x80 to 0x9F.

The **TSP** character (0x20) represents a transparent space. This character has no text foreground or background color; i.e., it passes through the fill color of the window containing it.

The **NBTSP** character (0x21) represents a non-breaking transparent space. When *wordwrap* = 0, this character is the same as a transparent space (TSP). Interpretation of this code when *wordwrap* = 1, is reserved.

The █ character (0x30) is a solid block which fills the entire character position with the text foreground color.

Also included in this set are the block-drawing characters, opening and closing single and double quote marks, the trade mark symbol, the service mark symbol, and the remaining Latin-1 characters.

b3-b0 \ b7-b4	2	3	4	5	6	7
	G2					
0	TSP	█				
1	NBTSP	‘				
2		’				
3		“				
4		”				
5	...	•				
6						⅛
7						⅜
8						⅝
9		™				⅞
A	Š	š				│
B						┐
C	Œ	œ				└
D		℠				─
E						┘
F		Ÿ				┌

Table 14 G2 Code Set

The Code 0x13 (EXT1) at the start of a syntactic element, followed by a code from the G2 Code Set indicates that the two bytes form a two-byte syntactic element.

7.1.8.1 Display of Undefined G2 Code Set Characters

When encountering an undefined character in the G2 Code Set, decoders should display either a space (0x20) or an underline character (0x5F). See Section 7.1.6.

7.1.9 G3 Code Set - Future Expansion

The G3 code set currently contains the single caption icon CC . The remainder of the code set is reserved for future expansion. See Table 9.

Characters in the G3 space are transmitted by preceding them with the control character EXT1 (0x10) from the C0 code set. Following the EXT1 extender prefix, the G3 characters are addressed with the base codes in the range 0xA0 - 0xFF.

Decoders should display the underline character (0x5F - G0 Code set) when encountering unsupported graphic symbols in the G3 Code set. See Section 7.1.7.

b3-b0 \ b7-b4	A	B	C	D	E	F
0	CC					
1						
2						
3						
4						
5						
6						
7						
8						
9						
A						
B						
C						
D						
E						
F						

Table 15 G3 Code Set

The Code 0x13 (EXT1) at the start of a syntactic element, followed by a code from the G3 Code Set indicates that the two bytes form a two-byte syntactic element.

7.1.10 C2 Code Set - Extended Control Code Set 1

The C2 code set is reserved for future extended miscellaneous control and caption command codes. Codes in the C2 space are transmitted by preceding them with the control character EXT1 (0x10) from the C0 code set. Following the EXT1 extender prefix, the C2 characters are addressed with the base codes in the range 0x00 – 0x1F.

These codes can be succeeded by additional data bytes per the following:

a) Codes 0x00 through 0x07 are single-byte control codes (0 - additional bytes).
b) Codes 0x08 through 0x0F are two-byte control codes (1 - additional byte).
c) Codes 0x10 through 0x17 are three-byte control codes (2 - additional bytes).
d) Codes 0x18 through 0x1F are four-byte control codes (3 - additional bytes).

Example: The total sequence for a four-byte control code would be:

EXT1, 0x18, <data1>, <data2>, <data3>

DTVCC decoders which do not implement these commands use their implied sizes to skip over them in the Service Blocks.

		C	2
	b7-b4	0	1
b3-b0	0		
	1		
	2		
	3		
	4		
	5		
	6		
	7		
	8		
	9		
	A		
	B		
	C		
	D		
	E		
	F		

Table 16 C2 Code Set

The Code 0x10 (EXT1) at the start of a syntactic element, followed by a Code in the range 0x00 through 0x07 indicates that the two bytes form a two-byte syntactic element.

The Code 0x10 (EXT1) at the start of a syntactic element, followed by a Code in the range 0x08 through 0x0F indicates that the two bytes, with the byte following them, form a three-byte syntactic element.

The Code 0x10 (EXT1) at the start of a syntactic element, followed by a Code in the range 0x10 through 0x17 indicates that the two bytes, with the two bytes following them, form a four-byte syntactic element.

The Code 0x10 (EXT1) at the start of a syntactic element, followed by a Code in the range 0x18 through 0x1F indicates that the two bytes, with the three bytes following them, form a five-byte syntactic element.

7.1.10.1 Use of EXT1

Table 17 lists the bytes that should be skipped by a decoder when encountering the undefined codes in Code Set C2.

Extended Code	Skip
0x00 – 0x07	EXT1,ExtCode
0x08 – 0x0F	EXT1,ExtCode,<data1>
0x10 – 0x17	EXT1,ExtCode,<data1>,<data2>
0x18 – 0x1F	EXT1,ExtCode,<data1>,<data2>,<data3>

Table 17 Extended Codes and Bytes to Skip—C2 Code Set

7.1.11 C3 Code Set - Extended Control Code Set 2

The C3 code set is an additional set reserved for future extended miscellaneous control and caption command codes. See Table 19. Codes in the C3 space are transmitted by preceding them with the control character EXT1 (0x10) from the C0 character set. Following the EXT1 extender prefix, the C3 command codes are addressed with the base codes in the range 0x80 – 0x9F.

7.1.11.1 Fixed Length Commands from 0x80 to 0x8F

Codes 0x80 through 0x8F are reserved for fixed-sized commands. These codes can be succeeded by additional data bytes per the following:

a) Codes 0x80 through 0x87 are five-byte control codes (4 - additional bytes).
b) Codes 0x88 through 0x8F are six-byte control codes (5 - additional bytes).

Example: The total sequence for a six-byte control code would be:

EXT1, 0x88, <data1>, <data2>, <data3>,<data4>,<data5>

DTVCC decoders which do not implement these commands shall use their implied sizes to skip over them in the Service Blocks. See Table 18.

Extended Code	Skip
0x80 – 0x87	EXT1,ExtCode,<data1>,<data2>,<data3>,<data4>
0x88 – 0x8F	EXT1,ExtCode,<data1>,<data2>,<data3>,<data4>,<data5>

Table 18 Extended Codes & Bytes to Skip—C3 Code Set

The Code 0x10 (EXT1) at the start of a syntactic element, followed by a Code in the range 0x80 through 0x87 indicates that the two bytes, with the four bytes following them, form a six-byte syntactic element.

The Code 0x10 (EXT1) at the start of a syntactic element, followed by a Code in the range 0x88 through 0x8F indicates that the two bytes, with the five bytes following them, form a seven-byte syntactic element.

7.1.11.2 Variable Length Codes from 0x90 to 0x9F
Codes 0x90 through 0x9F are reserved for variable-length caption commands. Variable-length caption commands have a 1-byte header following the command code. This header contains a 2-bit Type field (b7 - b6), a fixed zero bit (b5), which shall be set to '0', and a 5-bit Length field (b4 - b0). The Type field allows a command to be broken up into several segments. The Type field has the following possible values:

a) 00 - Beginning of Command (BOC),
b) 01 - Continuation of Command (COC), or
c) 11 - End of Command (EOC).

When a command is entirely contained in one segment, use binary 11 (EOC) to indicate that all of the command is in that segment.

The Length field ranges from 0 - 27 and indicates the number of data bytes following the header. The data bytes can have any 8-bit value. Only one variable-length caption command per service can be transmitted at a time, and the command segments shall be transmitted in order. This variable-length command capability is intended for commands which require the downloading of large units of data (e.g., fonts and graphics). DTVCC decoders which do not implement these commands should use the Length field to skip over them in the Service Blocks. See Table 19.

	C	3
b3-b0 \ b7-b4	8	9
0		
1		
2		
3		
4		
5		
6		
7		
8		
9		
A		
B		
C		
D		
E		
F		

Table 19 C3 Code Set

The code 0x10 (EXT1) at the start of a syntactic element, followed by a code in the range 0x90 through 0x9F indicates that these two code bytes, with a one byte header and the data bytes following the header (as defined in this section) form a syntactic element.

7.1.11.3 C3 Code Set Variable Length Commands- Encoding Guidelines

The variable-length caption command codes have been reserved for future expansion, and have no current use in DTV Closed Captioning. Decoders should handle this syntax by skipping over the contents of these variable-length commands; this provides compatibility with future enhanced DTV Closed Caption services.

7.1.11.3.1 Size of Variable Length Commands

Variable-length command segments should not span multiple Service Blocks. All associated command segments should lie within a single Service Block.

Because Service Block payloads are limited to 31 bytes, Command lengths shall be limited to 28 bytes (since the Service Block also holds the EXT1 byte, the Command code and the Header byte).

7.1.11.3.2 Decoding Variable Length Codes from 0x90 to 0x9F

When decoding variable-length codes from 0x90 to 0x9F, the Type field should be ignored and bytes should be skipped based on the Length field. If the Length is zero, no bytes should be skipped after the header. If the Length is one, one byte should be skipped after the header. See Table 20.

Extended Code	Skip
0x90 – 0x9F	EXT1,ExtCode,<data1(header)>,...<dataN> N = (data1 & 0x3F) +1

Table 20 Extended Codes and Bytes to Skip 0x90-0x9F

8 DTVCC Interpretation Layer

The DTVCC Interpretation Layer defines the DTVCC GUI. This discussion includes how the caption data coding is to be formatted when encoded and how it is to be interpreted when decoded. While the Caption Data Services Coding Layer (Section 7) identifies how service data bytes are represented, the Interpretation Layer describes how these bytes of data are to be processed. The data bytes for each caption service are interpreted as a unique data stream, independent from the other services.

This section defines all required and optional DTV closed-captioning features. Refer to Section 8.11 for a list of features required within a minimum DTVCC receiver implementation.

8.1 DTVCC Caption Components

The five major components of DTVCC captioning are Caption Screen, Caption Windows, Caption Pens, Caption Text, and Caption Synchronization.

a) **Caption Screen**—The Caption Screen is the canvas on which caption windows are displayed.

b) **Caption Windows**—The heart of caption definition consists of Caption Windows within which caption text is displayed.

c) **Caption Pens**—The Caption Pens component defines styles and attributes for the appearance of the text within the caption windows.

d) **Caption Text**—The Caption Text component defines how text is encoded and directed to specific windows.

e) **Caption Synchronization**—The Caption Synchronization component controls the flow of interpretation of commands and caption text within the independent service data streams.

The following subsections provide a general overview of these components. These subsections are then followed by a detailed presentation of the DTVCC caption command set.

8.2 Screen Coordinates

A set of coordinates is defined to map a rectangular grid onto the "safe-title" area of the screen. "Safe title" area shall be as defined in SMPTE RP 218. This grid is used to specify the position of caption windows.

Receivers that decode the DTV bitstream may have a 16:9, 4:3, or other display screen aspect ratio. The coordinate-system grid size for a 16:9 receiver is 210 horizontal cells by 75 vertical cells. The 4:3 coordinate-system grid size is 160 horizontal cells by 75 vertical cells.

The grid coordinates are specified as a pair of values in the form: (horizontal, vertical). The origin is the point in the upper left-most corner of the safe-title area, and is assigned the coordinate (0, 0). For the 16:9 format, the upper-right corner is (209, 0), the lower-left corner is (0, 74), and the lower right corner is (209, 74). A similar set of reference points is defined for the 4:3 format's 160 x 75 coordinate system: respectively, (0,0), (159, 0), (0, 74), and (159, 74). It is important to remember that these grid cells are not intended for text positioning, but for window positioning. Figure 12 is an example of a window and its grid location within the screen. The window and grid are assumed to extend beyond the right side and bottom of the section of grid as shown in Figure 12.

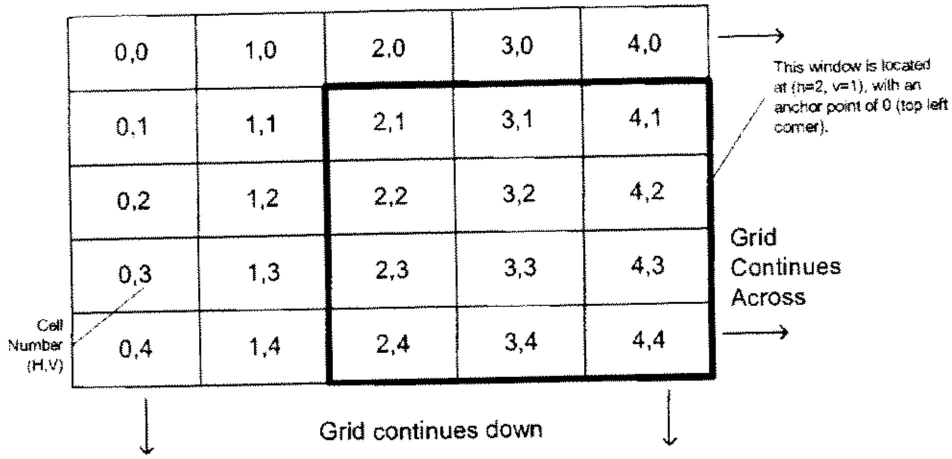

Figure 12 Example of Window and Grid Location

Once the window is positioned, the starting positioning of the text rows and columns follow, depending upon the size of the displayed font. The ending of the text rows depends upon the spacing (monospace vs. proportional space) of the chosen font.

Figure 13 shows an exaggerated relationship between a 16:9 screen, the 210 x 75 grid, the overscan area, the minimum viewable display area, the safe-title area, and the caption windows. The 4:3 screen is similar, but not shown.

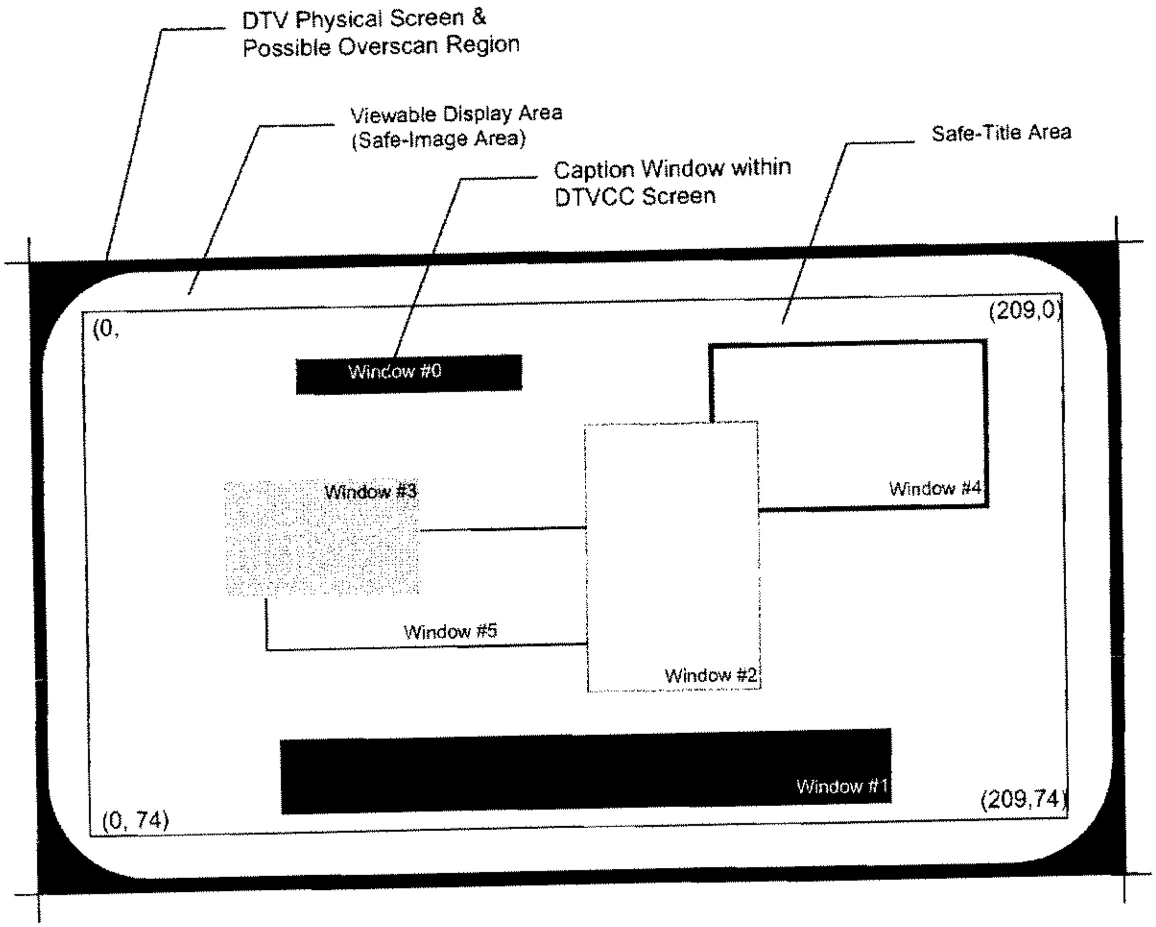

Figure 13 DTV 16:9 Screen and DTVCC Window Positioning Grid

8.3 User Options

Receiver manufacturers have the option to provide controls which may allow users to override styles and attributes specified in the service channel caption streams. Optional user controls might consist of caption font size, caption color and caption intensity (e.g. brightness) overrides. For further discussion, see the minimum DTVCC receiver decoder manufacturer recommendations in Section 8.11.

8.4 Caption Windows

All caption text is displayed and manipulated in receivers in the context of caption windows. There are 8 possible windows per service in which caption providers may write caption text. These windows may be implemented as buffers within receivers where any, none or all can be displayed at the same time. All 8 windows are available for the service currently selected by the user.

Manufacturers have the option of maintaining multiple sets of window buffers (i.e., instead of a single set of 8 buffers). Each set would be assigned to a service so that window processing of multiple services could occur simultaneously. This feature would have the effect that when a user switches services, the previously acquired and processed service data would be presented immediately. If only one set is used, the window buffers would have to be deallocated during service switching, and the new service would not appear until the buffers are reallocated and sufficient new service data are received.

The dimensions of a unique window specify an area on the screen which may contain caption text. The size of the window is based on the pen size (SMALL, STANDARD, or LARGE) that the user has selected. Caption text designated for the window may not exceed the boundaries of the window, regardless of the pen size the caption provider has specified or the pen size the user has chosen.

A window's size may change on screen when a user changes the pen size at the receiver. The effects of this window sizing are described further below.

8.4.1 Window Identifier

Each of the eight windows (and window buffers) is uniquely addressed by its window ID. Window ID numbers range from 0 to 7.

8.4.2 Window Priority

Each window has an associated priority which affects how it is displayed in conjunction with other windows that may be displayed at the same time. Priority 0 is the highest. Priority 7 is the lowest. When a higher priority displayed window overlaps a lower priority displayed window on the screen, the higher priority window shall appear in front of the lower priority window.
If windows of equal priority overlap (for any reason), caption decoding equipment may modify anchor points and pen attributes to make both windows fit within the safe title area. Decoders may also pick one or the other window to appear in front of the other. Implementations where all text in both windows is visible are preferred.

8.4.3 Anchor Points

There are 9 locations within a window which serve as "anchors". An anchor specifies the reference point for positioning the window on the screen, and the "shrink and grow" directions (see Figure 14) of the window and caption text within the window when a user changes the font size.

8.4.4 Anchor ID

The nine window anchor points shall be addressed by an Anchor ID which ranges from 0 to 8. Anchor ID points shall be identified as shown in Figure 14. Anchor ID 0 shall be located at the top-left corner of a window. Anchor ID 8 shall be located at the bottom-right corner of a window. Anchor ID 4 shall be located at the middle in the window.

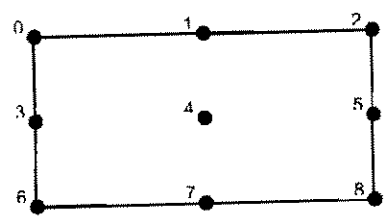

Figure 14 Anchor ID Location

Anchor points specify bounded and unbounded areas of caption text expansion and compression when the user overrides the standard font size for caption text display. Figure 15 shows the directions of expansion and compression of caption text for each anchor point. Solid lines indicate the bounded edges of the caption window. Dashed lines indicate the unbounded directions in which the caption text may shrink or grow.

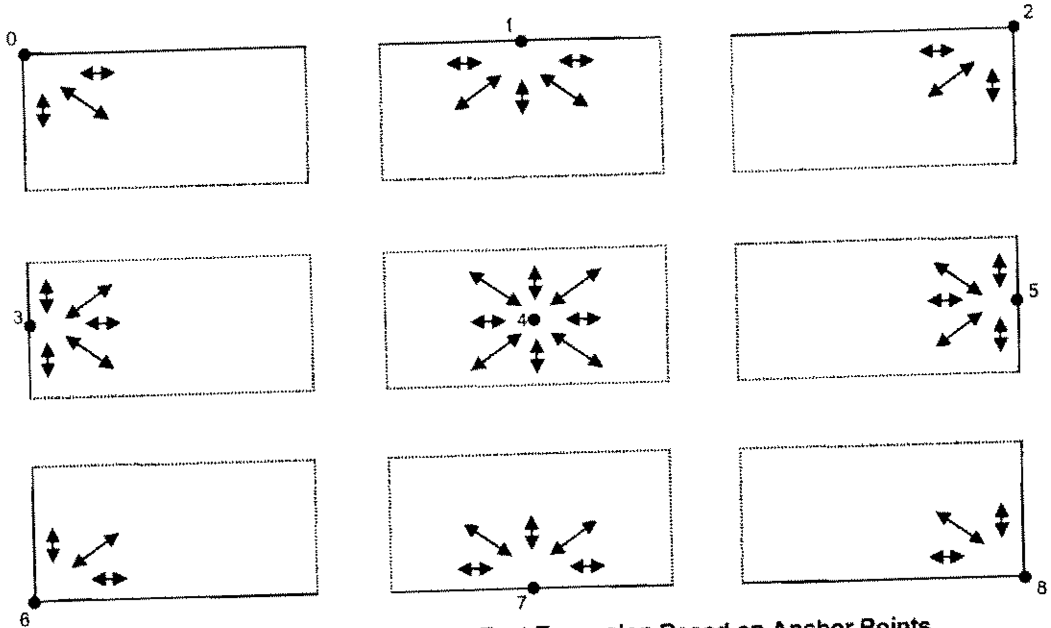

Figure 15 Implied Caption Text Expansion Based on Anchor Points

8.4.5 Anchor Location

The anchor location specifies where (in grid coordinates) the window's anchor point is to be physically located, and thus, the window itself. These grid coordinates are described in Section 8.2.

8.4.6 Window Size

The window size is specified in numbers of character rows and character columns for all display formats (16:9, 4:3, etc.). For all display formats, the maximum number of virtual rows shall be 15, and the maximum number of virtual columns shall be 32. The DefineWindow parameter values are zero-based. See Section 8.10.5.2.

As for the physical size of the window on the screen, the receiver scales the window based on the "effective font size". The effective font size is based on a combination of the pen size chosen by the caption provider and the font size chosen by the receiver user.

The method of determining the "effective font size" should be as follows:

a) If no font sizes are chosen by the user, the effective font size is the size of the font chosen by the caption provider.
b) If the user has chosen a substitute font, the effective font size is the size of the substitute font.

The height of the window is calculated as the number of rows multiplied by the physical height of the tallest character in the effective font size, to include line spacing. The width of the window is calculated as the number of columns multiplied by the physical width of the widest character in the effective font size, to include inter-character spacing.

8.4.6.1 Rows and Columns

The values of row count (rc) and column count (cc) are not the same as "character row count" and "character column count." In the DefineWindow command, rc and cc each specify a value which is the actual count minus 1.